Ethernet / シリアル通信

ワゴI/Oシステム

取り扱い説明書

Ver. 7.1 (2007.09.10)

＜　目　次　＞

1　ワゴ I/O システムの特長と概要 1-1
1.1　他のフィールドバスシステムへの変更 1-1
1.2　信頼性の高いケージクランプ接続 1-1
1.3　シンプルな取り付け、モジュールの変更追加がワンタッチ 1-1
1.4　ディジタル、アナログモジュールの豊富な品揃え 1-2
1.5　センサーへの電源供給ライン 1-3

2　ハードウェアの取扱い 2-1
2.1　シリアル通信バスカプラ　（７５０－３１２／３１４／３１５／３１６） 2-1
2.1.1　バスカプラの種類 2-1
2.1.2　バスカプラの取り扱いとモジュールのアセンブリ 2-1
2.1.3　ノード設定スイッチ 2-3
2.1.4　カップラ内部のスライドスイッチ（フィールドバスの終端） 2-3
2.1.5　ピンアサイン 2-4
2.1.6　パソコンとの接続 2-5
2.2　Ethernet　バスカプラ　（７５０－３４２／８４２） 2-6
2.2.1　ハードウェアの取り扱い 2-6
2.2.2　Ethernet の接続 2-7
2.2.3　LED の表示と動作状況 2-8
2.3　バスカプラ仕様書 2-8

3　通信のための設定 3-1
3.1　シリアル Modbus 通信バスカプラ 3-1
3.1.1　シリアル Modbus の通信について 3-1
3.2　通信用の内部 DIP スイッチの設定 3-1
3.2.1　ハイパーターミナル（ウィンドウズ）を使用するときの設定 3-4
3.3　Ethernet　バスカプラの設定（バスカプラに IP アドレスを書き込む） 3-5
3.3.1　BootP テーブルソフトウェア 3-5
3.3.2　Boot p テーブルの編集 3-6
3.3.3　BootP　Server の実行 3-7
3.3.4　IP アドレス設定の確認 3-8

4　ビット割当て 4-1
4.1　内部ビット幅 4-1
4.2　ビット割当優先順位 4-1
4.3　ビット割当の例 4-4

5　通信プロトコル 5-1
5.1　シリアル通信 Modbus プロトコル 5-1
5.2　データ読み出しと書き込みの機能分類 5-1
5.2.1　読み出し、書き込みコマンドの送出方法 5-3
5.2.2　ワゴ I/O からデータを読み出す 5-4
5.2.3　ワゴ I/O へデータを書き込む 5-7
5.2.4　（参考）CRC（巡回符号方式）コードの生成 5-13
5.3　Ethernet　Modbus／TCP プロトコル 5-18
5.3.1　Modbus/TCP　プロトコル 5-18
5.3.2　Modbus/TCP のデータ構造 5-18
5.3.3　データ読み出しと書き込みの機能分類 5-19
5.3.4　読み出し、書き込みコマンドの送出方法 5-20

5.3.5 ワゴ I/O からデータを読み出す 5-21
5.3.6 ワゴ I/O へデータを書き込む 5-24

6 標準添付 開発支援ソフト（Dynamic Link Library） 6-1
6.1 システムインストール 6-1
6.2 Ｃ言語および C++での使用について 6-2
6.3 ＤＬＬ（Wago_io.dll）の機能 6-2
6.3.1 「ＲＳ－２３２Ｃ，ＲＳ－４８５の場合」750-312, 750-315, 750-314, 750-316 6-2
6.3.2 「Ｅｔｈｅｒｎｅｔの場合」750-342, 750-842 6-2
6.4 サンプル プログラムについて (Sample.exe） 6-3
6.5 関数仕様書 6-5
6.5.1 モジュール構成（コンフィグレーション情報）通知に関する関数 6-5
6.5.2 ポートのオープン・クローズに関する関数 6-6
6.5.3 データ入力（取得）に関する関数 6-7
6.5.4 ディジタル出力に関する関数 6-10
6.5.5 アナログ出力に関する関数 6-11
6.5.6 カウンターモジュール取扱いの関数 6-12
6.5.7 コマンド回数確認のための関数 6-14
6.5.8 レジスタの読み書き関数 6-15
6.6 VB プログラムサンプル 6-16

7 デジタル入出力モジュール 7-1
7.1 デジタル入力（24V DC; 120V AC, 230V AC, 48V DC） 型番 750-400～415 7-1
7.2 8 チャンネル デジタル入力（DC 24V/3.0ms/0.2ms、正方向スイッチ） 型番 750-430, 431 7-4
7.3 4 チャンネル デジタル入力（DC 24V／3.0ms／0.5ms、2 線式） 型番 750-432, 433 7-6
7.4 8 チャンネル デジタル入力（DC 24V／3.0ms、負方向スイッチ） 型番 750-436 7-8
7.5 デジタル出力（標準） 型番 750-501～504, 516, 519 7-10
7.6 デジタル出力（診断機能付きの標準タイプ） 型番 750-506 7-12
7.7 8 チャンネル デジタル出力（DC24V／0.5A、正方向スイッチ） 型番 750-530 7-14
7.8 4 チャンネル デジタル出力（DC 24V／0.5A、2 線式） 型番 750-531 7-16
7.9 8 チャンネル デジタル出力（DC 24V／0.5A、負方向スイッチ） 型番 750-536 7-18
7.10 デジタル出力（ソリッドステートリレー） 型番 750-509 7-20
7.11 デジタル出力（リレー） 型番 750-512...514, 517 7-22

8 アナログ入力モジュール 8-1
8.1 2 チャンネルアナログ入力 0～20mA/4～20mA（外部電源利用型） 型番 750-452, 454 8-1
8.2 2 チャンネルアナログ入力 0～20mA/4～20mA （シングルエンド入力：システム電源供給型） 型番 750-465（12 ﾋﾞｯﾄ 0～20mA） 750-472（15 ﾋﾞｯﾄ 0～20mA） 750-466（12 ﾋﾞｯﾄ 4～20mA） 750-474（15 ﾋﾞｯﾄ 4～20mA） 8-4
8.3 2 チャンネルアナログ入力 +/-10V（差動、外部電源利用型） 型番 750-456（12 ﾋﾞｯﾄ） 8-8
8.4 2、4 チャンネルアナログ入力 0～10V（シングルエンド入力） 型番 750-467, 468 8-11
8.5 2 チャンネルアナログ入力 –10V～+10V/0～10V（15 ﾋﾞｯﾄ高分解能型） 型番 750-476,478 8-14
8.6 PT 100 用入力 型番 750-461 シリーズ 8-17
8.7 熱電対入力モジュール 型番 750-469（診断機能付） 8-20
8.8 4 チャンネルアナログ入力 0～20mA/4～20mA （シングルエンド入力：外部電源供給型） 型番 750-453, 455 8-25
8.9 4 チャンネルアナログ入力 ±10V/0-10V(シングルエンド入力) 型番 750-457,459 8-28
8.10 4 チャンネル RTD 用入力モジュール 型番 750-460 シリーズ 8-33
8.11 2 チャンネルアナログ入力 ±10V（差動、外部電源利用型） 型番 750-479 8-36
8.12 2 チャンネルアナログ入力 0-20mA（差動、外部電源利用型） 型番 750-480 8-39
8.13 2 チャンネルアナログ入力 4-20mA（絶縁差動、外部電源利用型） 型番 750-492 8-42

8.14 2 チャンネルアナログ入力モジュール AC/DC 0-1A（差動、外部電源利用型） 型番 750-475 8-45
8.15 2 チャンネルアナログ入力モジュール AC/DC 0-10V（差動、外部電源利用型） 型番 750-477..... 8-48
8.16 2 チャンネルアナログ入力モジュール 0-30V（差動、外部電源利用型） 型番 750-483 8-51

9 アナログ出力モジュール.....9-1
9.1 2 チャンネルアナログ出力 0～10V 型番 750-550.....9-1
9.2 2 チャンネルアナログ出力 0～20mA/4～20mA 型番 750-552, 554 9-4
9.3 2 チャンネルアナログ出力+/-10V 型番 750-556.....9-7
9.4 4 チャンネルアナログ出力 ±10V/0-10V 型番 750-557,559..... 9-10
9.5 4 チャンネルアナログ出力 0～20mA/4～20mA 型番 750-553, 555 9-14

10 カウンタモジュール 10-1
10.1 型番 750-404, 750-404/000-001, 750-404/000-002 アップ／ダウンカウンタ 100kHz, 750-40410-1
10.2 カウンタモジュール 型番 750-638 2ch アップ／ダウンカウンター 最大 500Hz 10-8

11 その他のモジュール 11-1
11.1 終端モジュール、電源拡張モジュール 型番 750-600, 750-614.....11-1
11.2 セパレーションモジュール 型番 750-616, 750-616/030-000, 750-621.....11-2
11.3 電源入力モジュール 型番 750-601, 602, 609, 610, 611, 612, 613, 615.....11-3
11.4 シリアルインタフェースモジュール RS232C インタフェース 型番 750-650（9600bps、ﾊﾟﾘﾃｨ無、8ﾋﾞｯﾄ、ｽﾄｯﾌﾟﾋﾞｯﾄ 1） 750-650/003-000（通信条件可変タイプ） RS485 インタフェース 型番 750-653（9600bps、ﾊﾟﾘﾃｨ無、8ﾋﾞｯﾄ、ｽﾄｯﾌﾟﾋﾞｯﾄ 1） 750-653/003-000（通信条件可変タイプ） 11-5

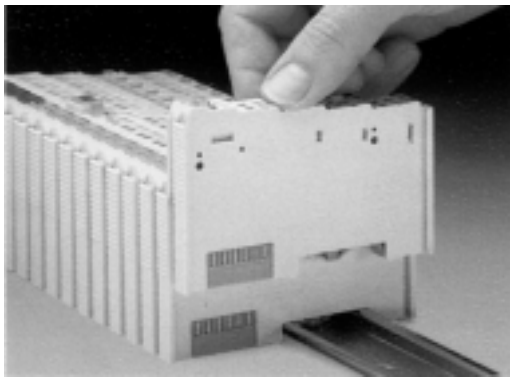
12 LabVIEW 開発用ソフトウェアライブラリ 12-1
12.1 ファイル構成..... 12-1
12.1.1 WAGO_IO.dll 呼び出し用 DLL（WdllCall.dll） 12-1
12.1.2 WagoIO.llb 概要..... 12-2
12.1.3 WagoSamples.llb 概要 12-3
12.2 基本命令－DLL（WAGO_IO.dll）呼び出し用 VI 12-3
12.2.1 DLL（WAGO_IO.dll）呼び出し用 VI 関数仕様書..... 12-4
12.3 目的別命令－WagoIO 入出力 VI 12-22
12.3.1 WagoIO 入出力 VI 関数仕様書..... 12-23
12.4 サンプルライブラリ解説 12-41
12.4.1 サンプル名：WagoPnlConfig.vi 12-43
12.4.2 サンプル名：Wago_AI_Sample.vi 12-47
12.4.3 サンプル名：Wago_AO_Sample.vi..... 12-49
12.4.4 サンプル名：Wago_DI_Sample.vi..... 12-51
12.4.5 サンプル名：Wago_DO_Sample.vi 12-53
12.5 データ収集アプリケーション例 （DAQ_Apply.vi） 12-55

13 LabVIEW 実行ソフトウェアサンプル.....13-1
13.1 データ収集システム DAQ_Sample1～４ 13-2

1 ワゴ I/0 システムの特長と概要

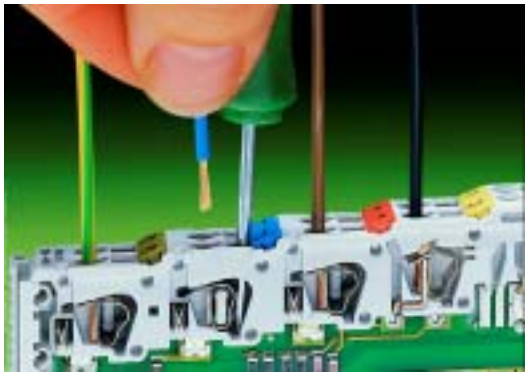
1.1 他のフィールドバスシステムへの変更

バスカップラー（通信制御モジュール）を含む各モジュールはスライド接続式のブロックになっており、将来異なるフィールドバスへ変更する場合でも、このバスカップラーを交換するだけで既存のオープンフィールドバスへの変更が可能です。このとき、センサーやアクチュエーターなどのフィールド機器を接続しているノード内のモジュール部分は変更する必要がなく、そのまま引き続き使用できます。



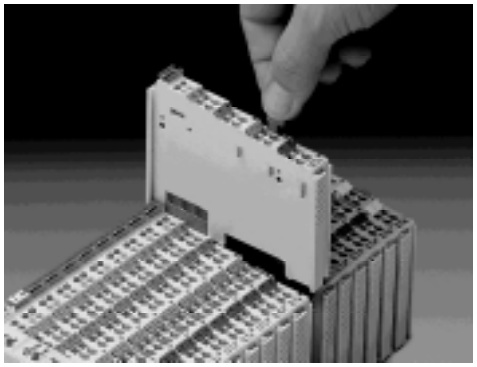
1.2 信頼性の高いケージクランプ接続

ワゴ I/O システムはセンサーやアクチュエータを直接結線できるケージクランプ接続方式を採用しております。スプリング方式によるこの接続方式は一旦接続したあとは、ネジ式端子台のようなメンテナンスが不要であるばかりではなく、センサー等の結線変更がワンタッチで実行できます。



1.3 シンプルな取り付け、モジュールの変更追加がワンタッチ

センサーやアクチュエータの組み合わせ変更が発生した場合でも、工具を使わずに簡単にモジュールの追加・削除、組替えができます。



1.4　ディジタル、アナログモジュールの豊富な品揃え

入出力モジュールにはアナログ、デジタル、熱電対など温度センサーを接続できるもの、さらにはカレントループやＲＳ４８５へ入出力できるインターフェイスモジュールなどを豊富に取り揃えております。しかもこれらの組み合わせは自由自在で、入出力モジュールの混在、アナログとデジタルモジュールの混在も可能です。

＜モジュール一覧＞　　2007/09/10.現在

750－342	Ethernet　10 BaseT	750－312	(RS-485)　19.2Kbps
750－842	Ethernet　10 BaseT	750－314	(RS-232C)　19.2Kbps
		750－315	(RS-485)　115.2Kbps
		750－316	(RS-232C)　115.2Kbps

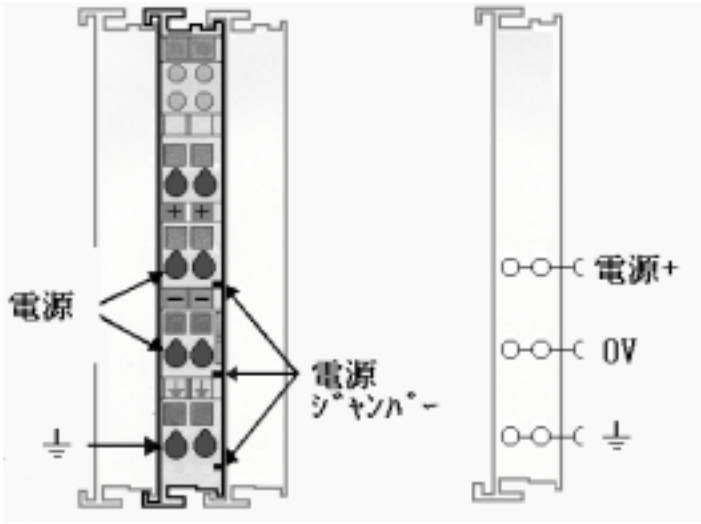
750－400	2 Ch ディジタル入力 DC24V、応答3ms	750－501	2 Ch ディジタル出力 DC24V 0.5A
750－401	2 Ch ディジタル入力 DC24V、応答0.2ms	750－502	2 Ch ディジタル出力 DC24V 2A
750－402	4 Ch ディジタル入力 DC24V、応答3ms	750－504	4 Ch ディジタル出力 DC24V 0.5A
750－403	4 Ch ディジタル入力 DC24V、応答0.2ms	750－506	2 Ch ディジタル出力 DC24V 0.5A診断機能付
750－405	2 Ch ディジタル入力 AC230V、応答10ms	750－509	2 Ch ソリッドステートリレー出力 AC230V、 300mA
750－406	2 Ch ディジタル入力 AC120V、応答10ms	750－516	4 Ch ディジタル出力 DC24V 0.5A 負方向接続
750－408	4 Ch ディジタル入力 DC24V、応答3ms、負方向	750－519	4 Ch ディジタル出力 DC5V 20mA
750－409	4 Ch ディジタル入力 DC24V、応答0.2ms、負方向	750－530	8 Ch ディジタル出力 DC24V 0.5A
750－410	2 Ch ディジタル入力 DC24V 応答3ms、センサー入力	750－531	4 Ch デジタル出力 DC24V 0.5A 2線式
750－412	2 Ch ディジタル入力 DC48V 応答3ms	750－536	8 Ch ディジタル出力 DC24V 0.5A 負方向接続
750－414	4 Ch ディジタル入力 DC5V 応答0.2ms	750－512	2 Ch リレー出力 最大AC230V 2A、＋側コモン a接点
750－415	4 Ch ディジタル入力 AC/DC24Vﾀﾞｲﾚｸﾄ入力	750－513	2 Ch リレー出力 最大AC230V 2A、独立型、 a接点
750－430	8 Ch ディジタル入力 DC24V、応答3ms	750－514	2 Ch リレー出力 最大 125V 0.5A、独立型、c 接点
750－431	8 Ch ディジタル入力 DC24V、応答0.2ms	750－517	2 Ch リレー出力 最大 250V 1A、独立型、c 接点
750－432	4 Ch ディジタル入力 DC24V、応答3ms、2線式		
750－433	4 Ch ディジタル入力 DC24V、応答0.2ms、2線式		
750－436	8 Ch ディジタル入力 DC24V、応答3ms、負方向		
750－404	1 Ch アップダウンカウンター DC24V、 100kHz		
750－638	2 Ch アップダウンカウンター CD24V, 500Hz		

750－452	2 Ch アナログ電流入力、 0-20mA	750－550	2 Ch アナログ電圧出力 0-10V、
750－453	4 Ch アナログ電流入力、0-20mA、外部電源利用型	750－552	2 Ch アナログ電流出力 0-20mA、
750－454	2 Ch アナログ電流入力、 4-20mA	750－554	2 Ch アナログ電流出力 4-20mA、
750－455	4 Ch アナログ電流入力、4-20mA、外部電源利用型	750－556	2 Ch アナログ電圧出力 ±10V、
750－456	2 Ch アナログ電圧入力、 ±10V、 差動入力型	750－557	4 Ch アナログ電圧出力 ±10V
750－457	4 Ch アナログ電圧入力、 ±10V、シングルエンド	750－559	4 Ch アナログ電圧出力 0-10V
750－459	4 Ch アナログ電圧入力、0-10V、シングルエンド		
750－460	4 Ch RTDセンサー入力 各種、2線用		
750－461	2 Ch RTDセンサー入力 各種		
750－462	2 Ch 熱電対入力 各種		
750－465	2 Ch アナログ電流入力、 0-20mA、電源供給型		
750－466	2 Ch アナログ電流入力、 4-20mA、電源供給型		
750－467	2 Ch アナログ電圧入力 0-10V、シングルエンド		
750－468	4 Ch アナログ電圧入力 0-10V、シングルエンド		
750－469	2 Ch 熱電対入力、診断機能付		
750－472	2 Ch アナログ電流入力、 0-20mA 高分解能(15ビット)		
750－474	2 Ch アナログ電流入力、 4-20mA 高分解能(15ビット)		
750－475	2 Ch アナログ電流入力 AC/DC 0-1A		
750－476	2 Ch アナログ電圧入力、 ±10V、 高分解能(15ビット)		
750－477	2 Ch アナログ電圧入力、 AC/DC 0-10V 差動入力型		
750－478	2 Ch アナログ電圧入力、 0-10V、 高分解能(15ビット)		
750－479	2 Ch アナログ電圧入力、 ±10V、 差動入力型		
750－480	2 Ch アナログ電流入力、 0-20mA 差動入力型		
750－483	2 Ch アナログ電圧入力、 0-30V 差動入力型		
750－492	2 Ch アナログ電流入力、 4-20mA 絶縁差動入力型		

750－650	RS232Cインタフェース、パリティ無 8ビット ストップビット1	750－601	電源モジュール ヒューズ付DC24V 6.3A
750－650 /003－000	RS232Cインタフェース、通信条件可変タイプ	750－602	電源モジュール DC24V
		750－609	電源モジュール ヒューズ付AC230V 6.3A
750－653	RS485インタフェース、パリティ無 8ビット ストップビット1	750－610	電源モジュール ヒューズ付DC24V 6.3A、 診断機能付
750－653 /003－000	RS485インタフェース、通信条件可変タイプ	750－611	電源モジュール ヒューズ付AC230V 6.3A、 診断機能付
		750－612	電源モジュール AC、DC 0～230V ヒューズなし
750－600	終端モジュール	750－613	電源モジュール システムおよびフィールド用DC24V
		750－614	電源端子拡張モジュール
		750－615	電源モジュール ヒューズ付AC120V 6.3A

1.5 センサーへの電源供給ライン

各モジュールに接続するセンサーやアクチュエータに対しては、ワゴ I/O から電源を供給することが可能です。単独のノード内で複数種類の供給電源を設定するには、各セクションごとに電源供給モジュールを挿入して DC 、AC の各電圧に対応した電源供給ラインを準備してください。



2　ハードウェアの取扱い

2.1　シリアル通信バスカプラ　（７５０－３１２／３１４／３１５／３１６）

2.1.1　　バスカプラの種類

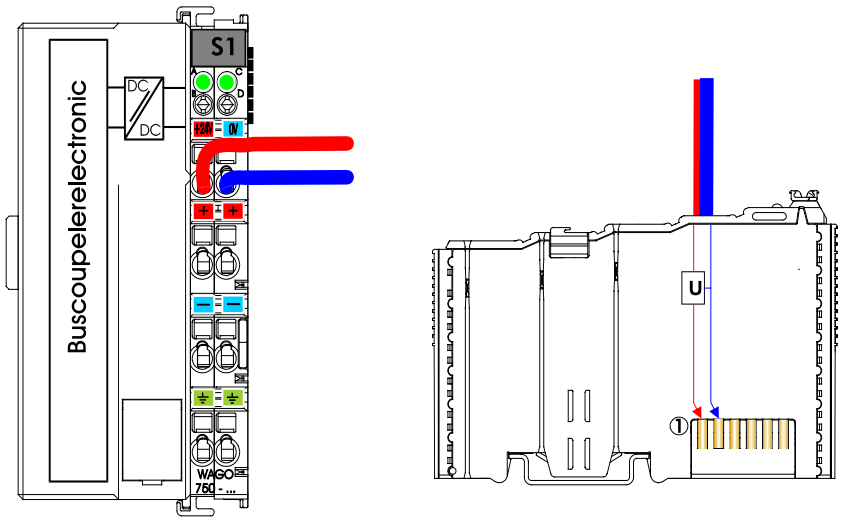
シリアル通信にて外部と通信できるユニットをバスカプラと呼びます。シリアル通信仕様のバスカプラは以下の４種類のタイプがあります。尚、通信プロトコルについては MODBUS 仕様に基づいています。MODBUS は公開された一般的な通信仕様で、MODBUS プロトコルを採用したコントローラに直接接続して通信することもできますが、パソコンやその他のシリアル通信機器でも簡単に通信制御することが可能です。（Modbus 通信プロトコルについては第５章で詳しく説明しています。）

型名	仕　　様
７５０－３１２	ＲＳ４８５　準拠、　最大通信速度　１９．２　ｋ　Ｂａｕｄ
７５０－３１４	ＲＳ２３２Ｃ準拠　　最大通信速度　１９．２　ｋ　Ｂａｕｄ
７５０－３１５	ＲＳ４８５　準拠　　最大通信速度　１１５．２ｋ　Ｂａｕｄ
７５０－３１６	ＲＳ２３２Ｃ準拠　　最大通信速度　１１５．２ｋ　Ｂａｕｄ

2.1.2　　バスカプラの取り扱いとモジュールのアセンブリ

＜供給電源（駆動電源）＞

バスカプラを駆動させるためには、24VDC の電源が必要となります。電源は図のようにカプラ最上部にある最初の 2 個のケージクランプに接続します。24V DC の電源電圧は内部の電圧レギュレータ（DC/DC コンバータ）によって変換され、この電圧が電子回路部（5V DC）に供給されます。外部バスシステムに対する電気的絶縁は、光カップラーによって行われます。尚この５VDC はバスカプラに接続された各機能モジュールを駆動する電源としても使用され、内部の「Ｋ－バス」を通じて供給されます。



各機能モジュール内部に実装されている制御用の電子回路部の電源結線は、モジュールをアセンブリ上に嵌め込むと次のモジュールのデータ接点によって自動的に行われ、付加された I/O モジュールへの電源は、メッキが施されたセルフクリーニングのスライド式接点を通して供給されます。既に設定済みのコンフィギュレーションから付加モジュールを取り出すと、Ｋ－バスの通信は断たれます。バスカプラはこれを検出する能力を備えています。

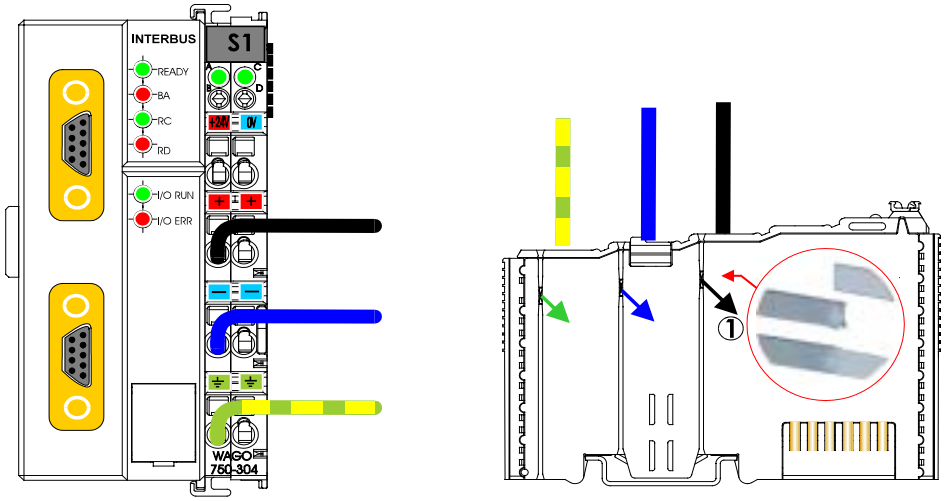
＜終端モジュール＞

バスカプラに各機能モジュールを接続した最終端には必ず７５０－６００（終端モジュール）を嵌め込んでください。Ｋ－バス用の終端モジュールです。

＜センサー用電源（フィールド側電源）＞

使用する各機能モジュールを一緒に DIN レール上に嵌め込むと、フィールド用電源電圧が自動的に供給されます。セルフクリーニングの電源ジャンパー接点（P. J. C.）があるので、安全な結線が保証されています。バスカップラーと I/O 筐体にはメス型接点が組込まれています。各機能モジュール（I/O）筐体にはオス型接点も同様に組込まれていますので、各 I/O モジュールを右方向に並べて挿入していけば、このオス型接点を通して各モジュールに電源電圧が供給されるように結線されます。

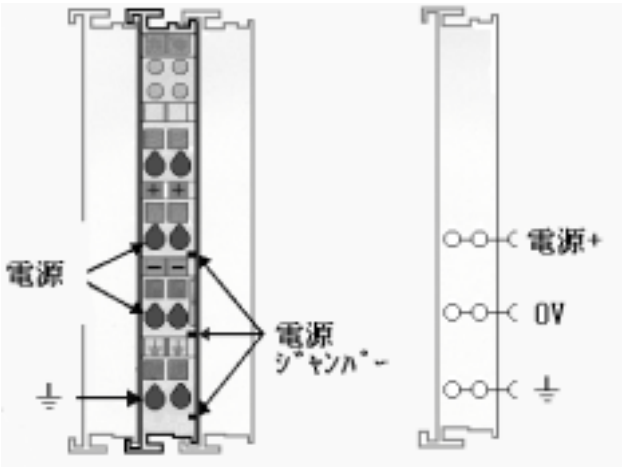
尚、バスカプラからセンサ用電源として供給できるのは 24VDC（最大 10A）に限定されています。これ以外の電源電圧が必要になる場合、電流が 10A 以上になる場合には７５０－６０９、７５０－６１２などの電源供給モジュールを別途お使いください。



＜電源モジュールの活用（フィールド側電源）＞

I/O 機能の内容に応じて、一部のモジュールには電源ジャンパ接点が用意されていないタイプがあります。アセンブリ時には、この点に注意してください。フィールド側の電源が必要のないモジュールはフィールド用電源ジャンパは用意されておらず、その後にフィールド電源が必要な I/O モジュールを追加接続していくためには、その前に電源モジュールを追加しなければなりません。また、単独のノード内で複数種類の供給電源を設定する場合、同じ供給電源仕様でもそのセクション内の総消費電流合計が、電源モジュールの容量をオーバーする場合にも、各セクションごとに電源供給モジュールを挿入して DC 、AC の各電圧に対応し電源供給ラインを準備してください。

図２－２－５



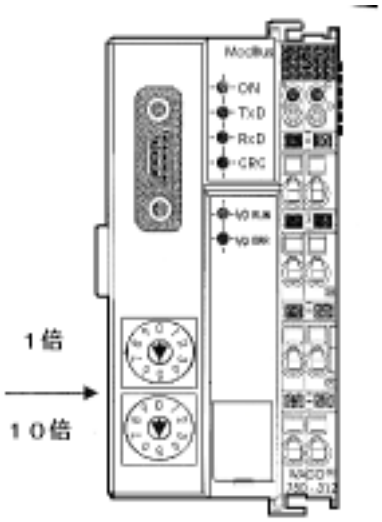
＜グラウンド（アース）＞

グラウンド（アース）接点は電気規格に準拠してファースト・メイク／ラスト・ブレークの動作構造になっており、内部回路保護用のアースとして使用できます。

2.1.3　　ノード設定スイッチ

バスカップラごとにノードアドレスロータリスイッチにて設定します。図で下側の数値は×１０倍、上側の数値は×１倍で００～９９（１０進法）の範囲で設定できます。１６進数では００～６３までの９９個のアドレスになります。

図２－３－５



2.1.4　　カップラ内部のスライドスイッチ（フィールドバスの終端）

ＲＳ４８５対応のバスカップラ、すなわち７５０－３１２および７５０－３１５は４線式(全二重通信)のＲＳ４８５に対応できます。通常、ＲＳ４８５バスラインにおいては両端の機器あるいはノードにターミネータ（終端抵抗）を付ける必要がありますが、これをワゴ I/O 内部のスライドスイッチにて設定することができます。

＜ハウジングのはずしかた＞

①バスカップラからすべての接続をはずして単独にします。
②ハウジングは底の金属板を両脇から抱え込むように小さなストッパー（突起）で留まっています。
③ハウジングを親指で横に広げるように押すと同時に、はずす方向へも力を加えます。
④ストッパーは２ヶ所あるので交互に両サイドに対して同じ動作を行います。
⑤ストッパーの部分が金属板部分からはずれたら、ゆっくりとハウジングをはずします。

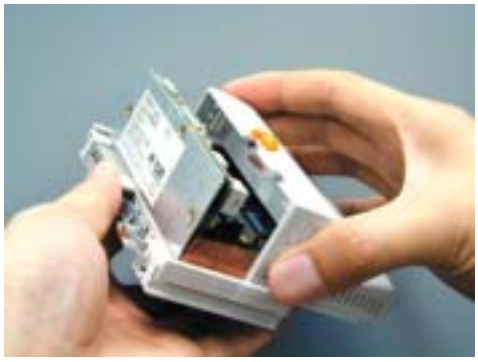
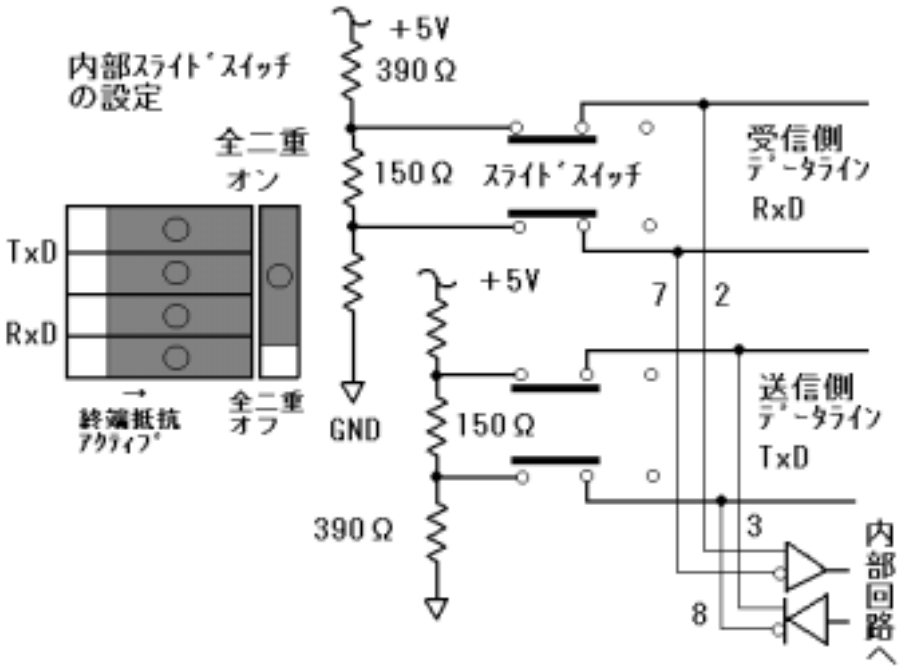


図２－４－１
RS485 バスラインの終端抵抗の設定



2.1.5　　ピンアサイン

ＲＳ４８５タイプ、すなわち７５０－３１２および７５０－３１５の D-sub９ピンコネクタのピンアサインは以下のとおりです。

ピン番号	信号名	入出力	内容
1			
2	RxD＋	入力	データ受信線（＋側）
3	TxD＋	出力	データ送信線（＋側）
4			
5			
6			
7	RxD－	入力	データ受信線（－側）
8	TxD－	出力	データ送信線（－側）
9			

（D-sub　９ピン　メス）

ＲＳ２３２Ｃタイプ、すなわち７５０－３１４および７５０－３１６のピンアサインは以下のとおりです。DOS-V パソコンのシリアルポートと接続する場合はストレートケーブルで接続します。

ピン番号	信号名	入出力	内容
1			
2	TxD	出力	データ送信線
3	RxD	入力	データ受信線
4			
5	ＧＮＤ		信号用グランド
6			
7			
8			
9			

（D-sub　９ピン　メス）

2.1.6　　パソコンとの接続

７５０―３１４、７５０－３１６の場合

750-314、750-316 と DOS-V パソコンの接続は以下のとおりです。**一般に市販されている「ストレートケーブル」にてそのまま接続できます。**

パソコン側
（ＤＯＳ－Ｖ）

信号名	ﾋﾟﾝ No
ＲｘＤ（受信）	2
ＴｘＤ（送信）	3
信号グランド	5

(D-sub　９ピンオス)

(D-sub9 ﾋﾟﾝﾒｽ)――(D-sub9 ﾋﾟﾝｵｽ)
ストレートケーブル

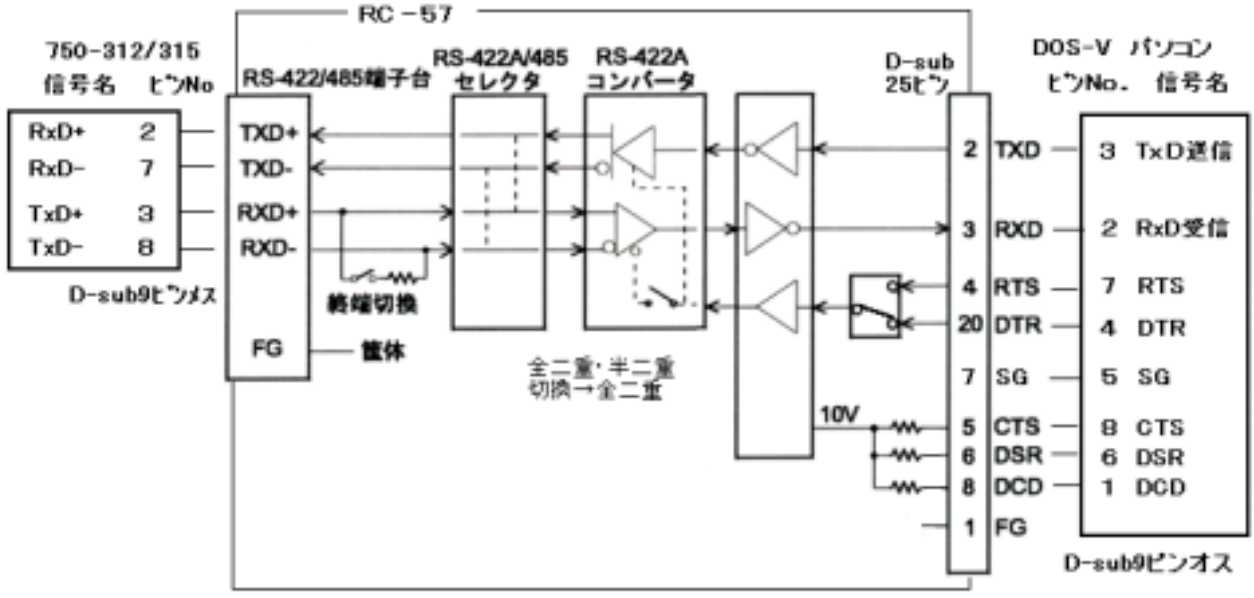
バスカップラー側
750-314、750-316

ﾋﾟﾝ No	信号名
2	ＴｘＤ（送信）
3	ＲｘＤ（受信）
5	信号グランド

(D-sub　９ピンメス)

７５０－３１２、７５０－３１５の場合

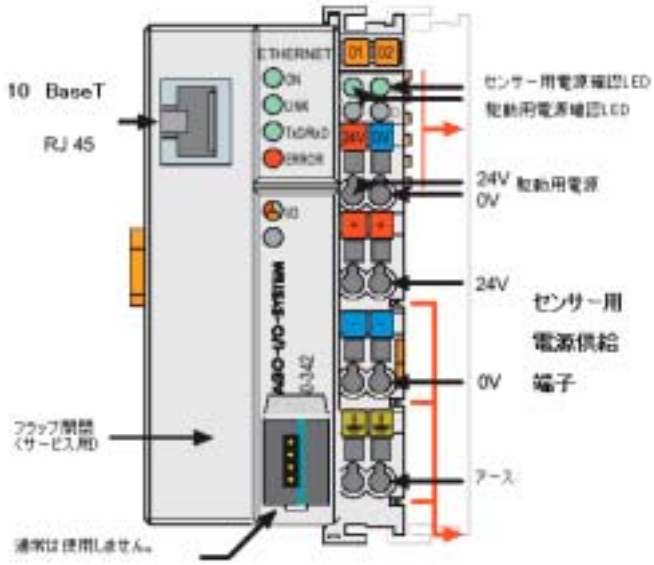
750-312 および 750-315 を RS-485 で接続する場合は、パソコン側のＲＳ－２３２ＣをＲＳ―４８５に変換するためのレベルコンバータが必要になります。レベルコンバータは４線式対応（全二重対応）のものをお使いください。半二重（２線式）には対応しておりません。以下の接続例は株式会社アール・エー・システムズ社のモデルＲＣ５７を使用した場合の接続例です。

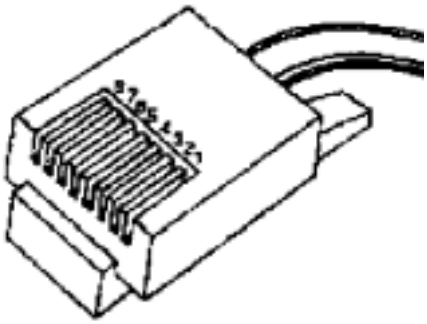


2.2　Ethernet　バスカプラ　（７５０－３４２／８４２）

2.2.1　　ハードウェアの取り扱い

① 駆動電源は DC24V です。この電圧は内部で DC-DC コンバータによって DC５V に変換されます。
② Ethernet 接続口は 10Base-T、RJ45 タイプの標準コネクタ対応です。
③ センサー用電源はジャンパーを介して次のモジュールへ供給されます。
④ フラップ開閉部のサービスポートはサービス用ですので使用しないでください。
⑤ RJ45 のコネクタ、およびピンアサインは一般使用のもので、以下のとおりです。





RJ－４５ピンアサイン

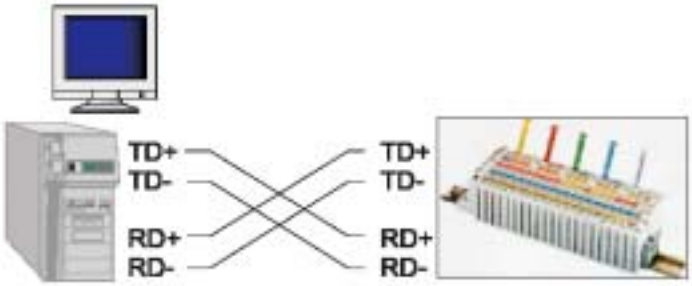
端子番号	信号名
1	TD＋　　送信＋
2	RD－　　送信―
3	RD＋　　受信＋
4	
5	
6	RD－　　受信―
7	
8	

2.2.2 Ethernet の接続

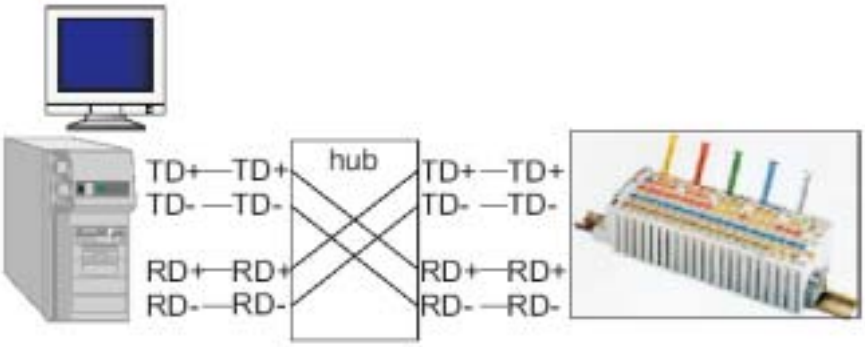
① バスカプラの接続部は 10Base-T の仕様に基づいて設計されています。ネットワーク構成詳細については 10Base-T の仕様に準拠します。

通信線	ツイストペア線
伝送方式	ベースバンド方式
速度	１０M bps
セグメントあたりのケーブル長	１００m
最大パケットサイズ	１５１２バイト
最小パケットサイズ	６４バイト

② パソコンとダイレクトに接続する場合はクロスタイプのケーブルを使用してください。

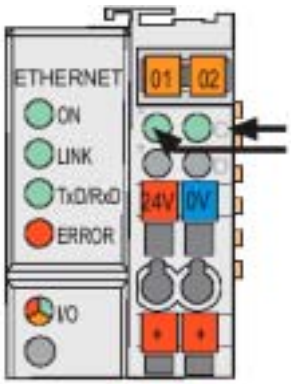


③ ハブ（Hub）を経由して接続する場合には通常のストレートケーブルを使用してください。



2.2.3 LED の表示と動作状況

表示 LED	表示内容
ON	バス（Ethernet）の初期化正常
LINK	物理的に接続完了
TxD/RxD	データ受信および送信
ERROR	バス（Ethernet）の接続エラー
I/O	I/O システム内の稼動状況 赤点滅は異常



2.3 バスカプラ仕様書

７５０－３４２仕様

接続 I/O モジュール数	最大６４
最大処理入力	最大５１２バイト
最大処理出力	最大５１２バイト
駆動電源	２４V　DC（-15%　～+20%）
内部最大消費電流	500 mA, 24V
内部入力電流	200mA,　5V
センサー用供給電源電圧および最大電流容量	２４V　DC（-15%　～+20%）１０A
動作温度	０℃　～　＋５５℃
保存温度	－２５℃　～　＋８５℃
電線接続	ケージクランプ方式 0.08～2.5ϕ（AWG28～AWG14） 被覆カット長　８～９　mm
外形寸法　W×H×L	５１×６５×１００
重量	約　１９５　g
振動、衝撃	IEC60068-2-6、　IEC60068-2-27
EMC	EN50082-2(95)、EN50081-2(94)
保護等級	IP20
その他認定規格	UL：　E175199 CE：　自己認定

3　通信のための設定

3.1　シリアル Modbus 通信バスカプラ

3.1.1　シリアル Modbus の通信について

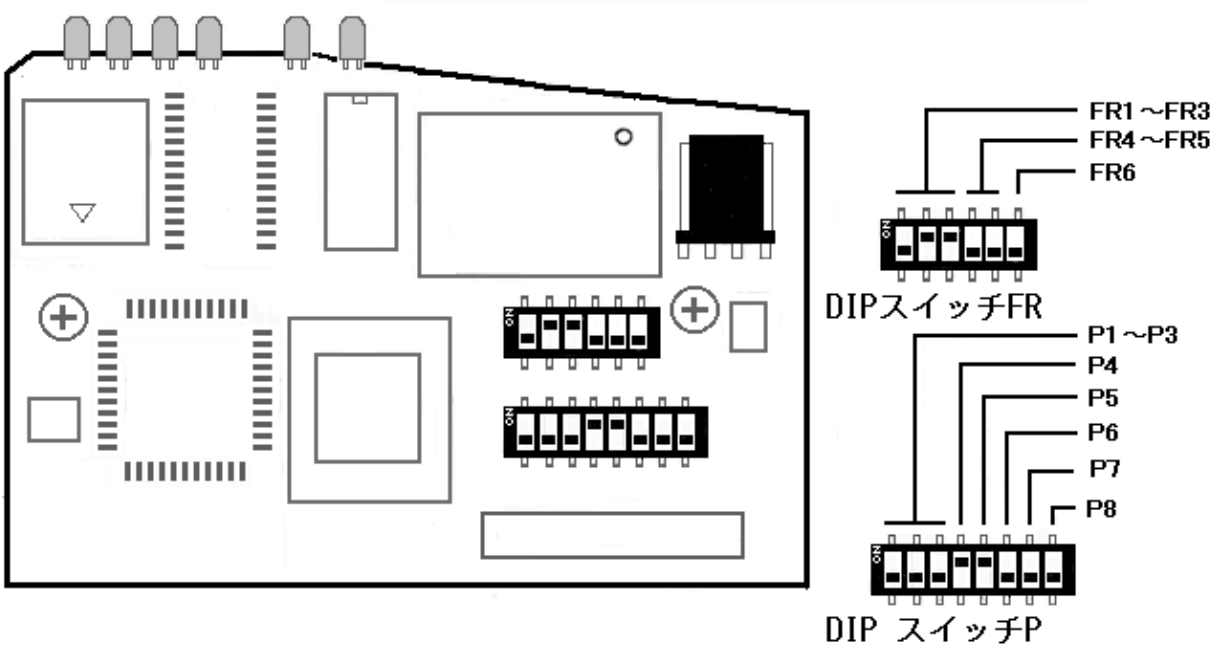
当社のシリアル通信ユニットはハードウェア通信仕様が RS232C および RS485、そして通信プロトコルはその仕様が公開されている Modbus に準拠しています。Modbus はシンプルな構造になっており、既に公開されていますので、RS232C や RS485 を装備した装置から通信・制御することが可能です。
Modbus の通信はマスター／スレーブにおける１対１の通信を基本としています。パソコン等のマスター側は毎回リクエスト信号を発信することによって動作指令を行います。このリクエスト信号データにはノードアドレス（スレーブのアドレス）が含まれており、これによって指定されたノードだけが動作を行ってレスポンス信号（アンサーバック）をマスター側へ返します。
ノード（スレーブ）側はリクエスト信号を受け取った時点で、リクエスト信号に含まれるノードアドレスを認識して自分に対するリクエストか否かを判断します。リクエスト信号の中にはノードアドレスの他に機能コードが含まれています。当該ノードはこの機能コードに従って動作を実行した後、レスポンス信号（アンサーバック）を返信します。マスター側はこのレスポンス信号を受け取ることによって指定したスレーブが正常に動作したかどうかを確認することができます。

送受信の信号の形式は ASCII モードと RTU モード（バイナリーモード）の２つの方法があります。
ASCII モードの場合は１バイト単位で構成された ASCII 文字コードによる通信です。 ASCII コードで通信する場合は、通信される文字データ間の時間間隔が一定でなくても送受信できます。
RTU モード（バイナリーモード）の場合には１バイトが２つの４ビット１６進数で構成されたバイナリーコードで送受信されます。同じボーレートならバイナリーコードの方が多くのデータを送受信できますが、データの通信は一定のスピードで行わなければなりません。

3.2　通信用の内部 DIP スイッチの設定

バスカップラ内部の DIP スイッチを起動前に設定をする必要があります。動作中に電源を入れたまま DIP スイッチを変更しても設定は変更されません。DIP スイッチを変更する場合は必ず電源を OFF にした状態で行ってください。その後、電源を入れ直してください。
DIP スイッチは図のように６ビットと８ビットの２つがあります。上側の６ビットスイッチを「DIP スイッチ FR」、下側の８ビットスイッチを「DIP スイッチＰ」と呼ぶことにします。

図３－２－１



① ボーレートの設定

ボーレートはDIP スイッチ FR の３ビット FR1，FR２，FR３で設定します。設定は以下の通りです。出荷時は９６００（baud）に設定されています。

ボーレート（baud）		FR1	FR2	FR3
７５０－３１２ ７５０－３１４	７５０－３１５ ７５０－３１６			
１５０	３８４００	off	off	off
３００	５７６００	on	off	off
６００	１１５２００	off	on	off
１２００	１２００	on	on	off
２４００	２４００	off	off	on
４８００	４８００	on	off	on
９６００	９６００	off	on	on
１９２００	１９２００	on	on	on

② エラー検出（データ長とパリティチェック）

送受信されるバイト単位の信号についていわゆるパリティビットを設定することができます。パリティビットは奇数パリティと偶数パリティをそれぞれ設定できますが、これはホスト側（パソコンなど）と同じ設定にしてください。出荷時は８ビット長、パリティなし、ストップビット１に設定されています。設定は以下のとおり DIP スイッチＦＲの３ビット FR４，FR５，FR６で設定します。尚、RTU モード（バイナリーモード）に設定されたときはデータ長が８ビットに固定されますので FR6 の設定は無視されます。

パリティ	データ長	ストップビット	FR4	FR5	FR6
パリティなし	８ビット	1	off	off	off
偶数パリティ	８ビット	1	on	off	off
奇数パリティ	８ビット	1	off	on	off
パリティなし	８ビット	2	on	on	off
パリティなし	７ビット	2	off	off	on
偶数パリティ	７ビット	1	on	off	on
奇数パリティ	７ビット	1	off	on	on
パリティなし	７ビット	3	on	on	on

③ データ転送モード

ASCII モードと RTU モード（バイナリーモード）の設定は DIP スイッチＰのビット４で設定します。出荷時は RTU モードに設定されています。

モード	Ｐ４
ASCII モード	off
RTU（ﾊﾞｲﾅﾘ）モード	on

④ データの終わり

RTU モード（バイナリーモード）において、通信フレームのデータの終わり（End Of File）を検出するための時間設定です。DIP スイッチＰの３ビット P1, P2, P3 で設定します。ASCII モードの時は関係ありません。

データの終わり	Ｐ１	Ｐ２	Ｐ３
３フレーム分	Off	off	off
１００　ｍ　秒	on	off	off
２００　ｍ　秒	off	on	off
５００　ｍ　秒	on	on	off
1　　秒	off	off	on
1　ｍ　秒	on	off	on
１０　ｍ　秒	off	on	on
５０　ｍ　秒	on	on	on

⑤ エラーチェックコード

ノード単位の動作テストなどを行う場合、エラーチェックコードを使わずに動作テストができます。DIP スイッチＰのビット５でエラーチェックコードを無視するかどうかを設定します。これはテスト用ですので、実際のアプリケーションにて駆動させるときには 必ず「実行」に設定しておいてください。初期設定は「実行」に設定されています。

エラーチェック	Ｐ５
無視	off
実行	on

⑥ その他

DIP スイッチ P のビット６ とビット７とビット８は必ず off に設定しておいてください。

P６	P７	Ｐ８
off	off	off

3.2.1　　ハイパーターミナル（ウィンドウズ）を使用するときの設定

ウィンドウズの標準アクセサリーとして付属している通信ソフト、「ハイパーターミナル」を使用して通信を行う時の設定について説明します。動作のチェックを文字列（アスキーコード）で送受信して動作確認するための設定内容です。

① 新しい接続を立ち上げたとき、「接続の設定」画面が出てきます。この画面で接続方法をシリアルポート、つまり COM1～COM４の中でシリアル通信ポートが設定されているコミュニケーションポートを選んでください。

② 次に「ポートの設定」を行います。ボーレート、データビット、パリティ、ストップビットの設定はバスカップラーと同じ設定にします。フロー制御については「なし」を選んでください。これらの内容を再度変更したいときは「プロパティ」をクリックして「モデムの設定」画面をクリックしてください。

③ バスカップラー側は ASCII モード、すなわちＰ４を「off」にしてください。またエラーチェックコードＰ５は無視「off」に設定してください。

④ 「プロパティ」をクリックして設定を選びます。ASCII の送信については、操作を行いやすくするために「行末に改行文字をつける」および「ローカルエコーする」のチェックボックスをそれぞれオンにしてください。また、ASCII の受信については「受信データに改行文字を付ける」および「右端で折り返す」のチェックボックスをオンにしてください。

⑤ DOS-V パソコンのシリアルポートと 750-314、750-316 の接続は以下のとおりです。**一般に市販されている「ストレートケーブル」にてそのまま接続できます。**

パソコン側
（ＤＯＳ－Ｖ）

信号名	ﾋﾟﾝ No
ＲｘＤ（受信）	2
ＴｘＤ（送信）	3
信号グランド	5

（D-sub　９ピンオス）

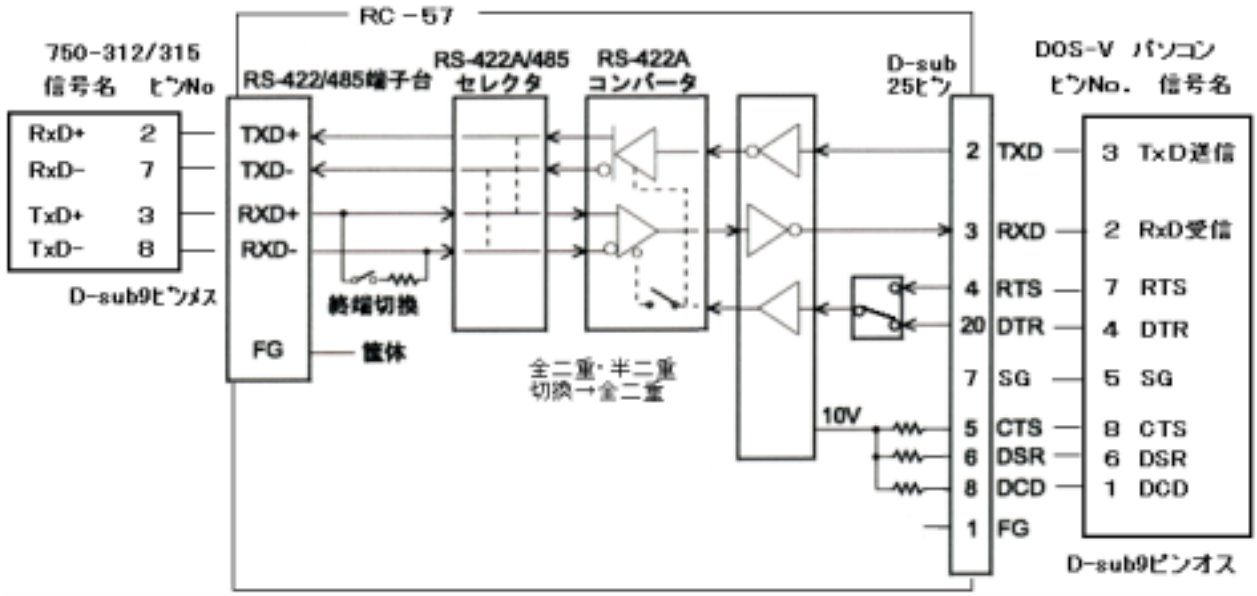
(D-sub9 ﾋﾟﾝﾒｽ)――(D-sub9 ﾋﾟﾝｵｽ)
ストレートケーブル

バスカップラー側
750-314、750-316

ﾋﾟﾝ No	信号名
2	ＴｘＤ（送信）
3	ＲｘＤ（受信）
5	信号グランド

（D-sub　９ピンメス）

⑥ 750-312 および 750-315 を RS-485 で接続する場合は、パソコン側のＲＳ－２３２ＣをＲＳ―４８５に変換するためのレベルコンバータが必要になります。レベルコンバータは４線式対応（全二重対応）のものをお使いください。半二重（２線式）には対応しておりません。以下の例は株式会社アール・エー・システムズ社のＲＣ―５７（絶縁型）を接続した場合の接続例です。



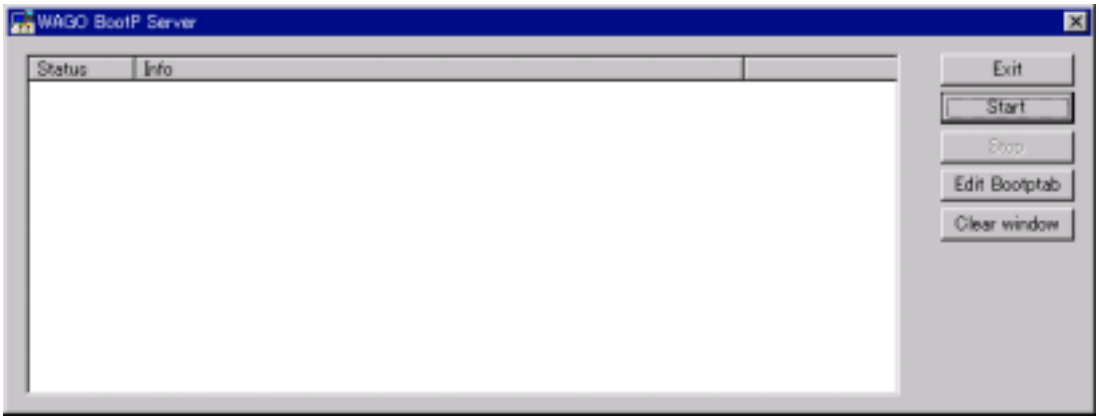
3.3 Ethernet　バスカプラの設定（バスカプラに IP アドレスを書き込む）

3.3.1 BootP テーブルソフトウェア

イーサネット対応バスカプラ（750-342、750-842）への IP アドレスを書き込みは BootP テーブルソフトにて行います。このソフトウェアはフリーウェアとして公開しており、下記 URL からダウンロードできます。ダウンロード後は、boot_p フォルダ内にある　setup.exe　をクリックすると自動的にインストールが実行されます。

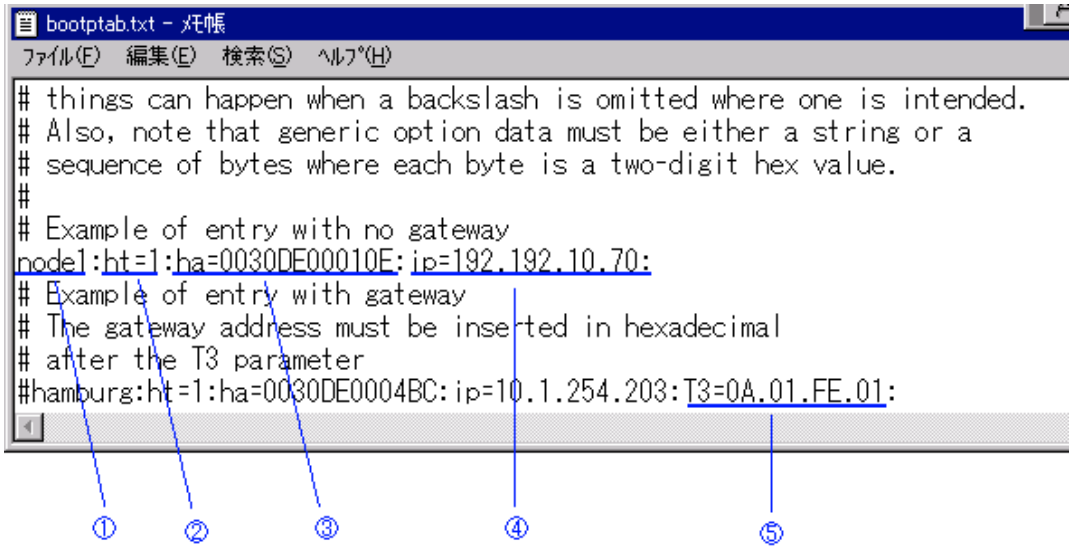
http://www.wago.com/wagoweb/japan/jap/index.htm

スタートメニュー／プログラム／Wago Software／WAGO Bootp Server／WAGO Bootp Server にてソフトウェアを立ち上げます。以下のような画面が表示されます。



3.3.2 Boot p テーブルの編集

Edit Bootptab をクリックすると、以下のように書込み用の Bootp テーブル（bootptab.txt）が表示されます。テーブル中の「#」マークは解説コメント用タグですので、その行は実行には関係ありません。Bootp テーブル中にはゲートウェイ無しの場合とゲートウェイがある場合の２種類のサンプル記述があります。



①はノード名を書込みます。半角のアルファベットと数字を使って設定したいノード名を記述できます。
②はハードウェアタイプを指定します。Ethernet のハードウェアタイプは　1 ですのでここは「**ht=1**」と記述してください。
③ここでは各々バスカプラに割り当てられた **MAC アドレス**を「**ha=**」に続けて記述します。バスカプラの **MAC アドレス**は各々のバスカプラ本体に記載されています。
④指定したバスカプラに書込みたい IP アドレスを「**ip= 」**に続けて記述します。
⑤バスカプラがゲートウェイの外にあるときは、経由したいゲートウェイのアドレスを **T3=** に続けて書込みます。
⑥書込みたい情報が記載してある行頭に付いている「#」をはずします。この状態で、このテキストファイルを上書き保存します。

<デフォルトゲートウェイ、サブネットマスクの設定>

サブネットマスク、デフォルトゲートウェイを設定する場合には ip アドレスに続いて以下のパラメータで設定してください。区切りは他の項目と同じくコロン（:）を入れてください。

sm=　小文字
gw=　小文字

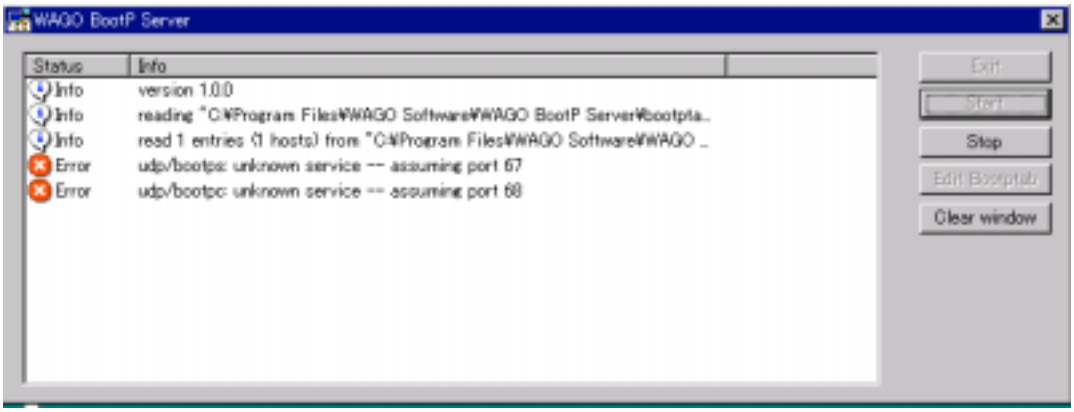
各々のパラメータに続いて、10 進数でタイプしたあと、上書き保存してください。

node1:ht=1:ha=0030DE00010E:ip=192.192.10.70:sm=255.255.255.0:gw=192.192.10.05

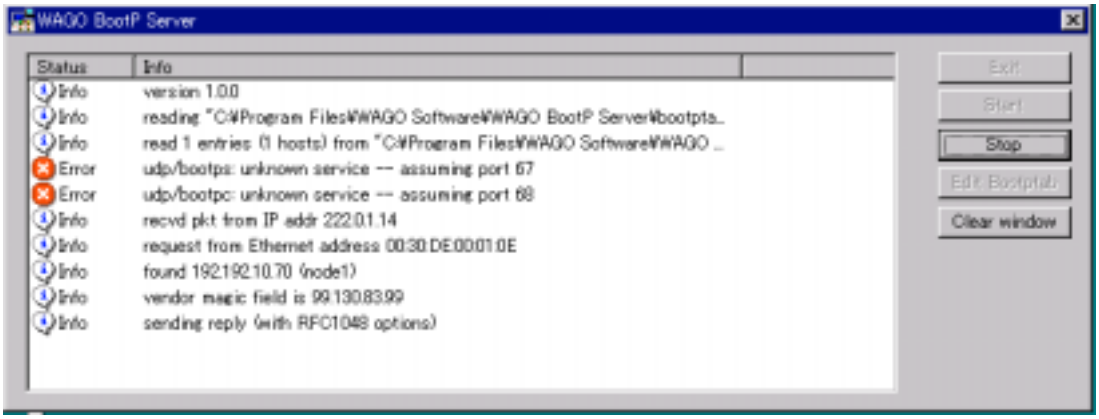
サブネットマスク　　ゲートウェイ

3.3.3 BootP Server の実行

① Bootp テーブルソフトの start ボタンを押します。そうすると BootP テーブルソフトウェアのウィンドウ中にエラーメッセージが表示されて、そこでストップします。



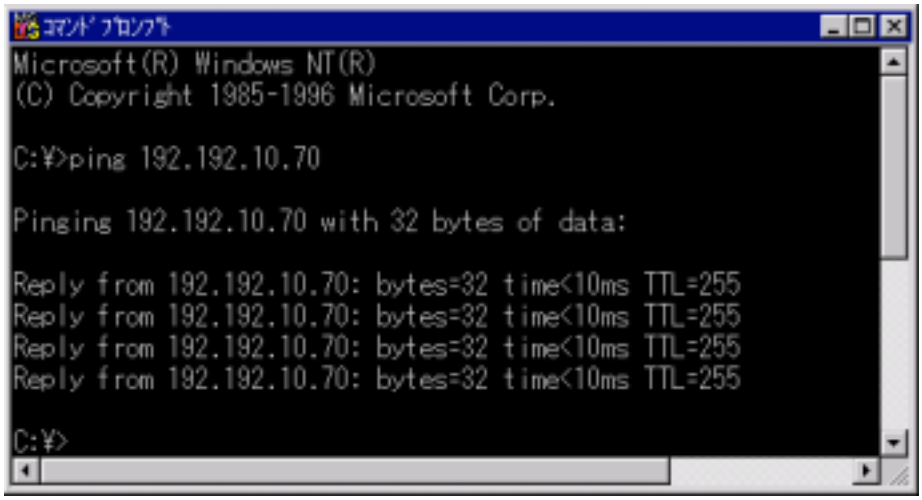
② ここでバスカプラの電源を一度切ります。
③ 電源を切断して２秒以上経過した後、再度電源を投入します。
④ 表示されていたエラーメッセージのあとに、設定した**ノード名**および**IP アドレス**が表示されます。
⑤ Stop を押してプログラムを終了させます。



3.3.4 IP アドレス設定の確認

① ウィンドウズのスタートメニューからプログラム中の「コマンド　プロンプト」を選択して実行し　ます。

② Dos の「ping」を実行します。ping コマンドに続いて、設定した IP アドレスを入力して実行し、以下のように応答が帰ってくるのを確認します。



③　ブラウザ（インターネットエクスプローラなど）を用いて直接設定を確認することができます。

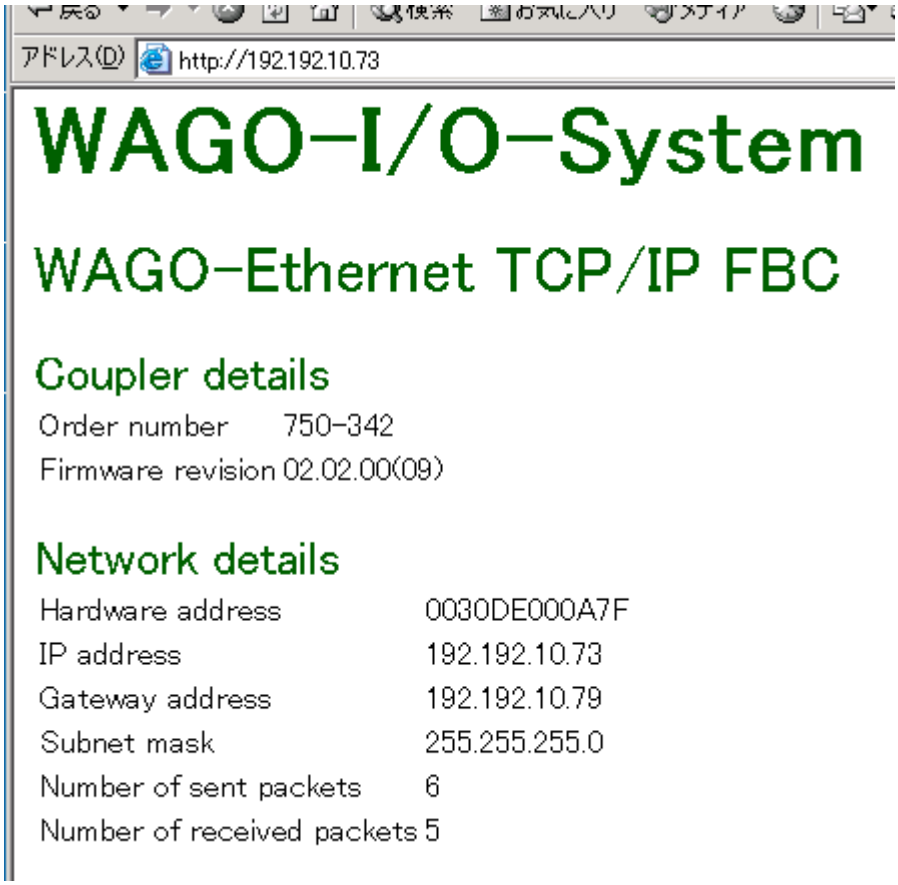
まず、ブラウザに「プロキシ」が設定されている場合はこれをはずします。プロキシをはずすには
インターネットエクスプローラのプルダウンメニューの「ツール」を選択して、

「インターネットオプション」→「接続」→「LAN の設定」でプロキシサーバの項目を表示したあと、
チェックボックスをはずしてください。

その後、アドレスの部分に IP アドレスをそのまま打ち込みます。

http://192.192.10.73

すると以下のようにバスカプラ内に設定された内容が HTTP ファイルで表示されます。



以上

4　ビット割当て

4.1　内部ビット幅

各々のモジュールは固有のビット幅（内部ビット幅）を持っています。この値は個々のモジュールによってその値が決まっており、原則的に入力モジュールは入力の内部ビット幅を、出力モジュールは出力の内部ビット幅を持っています。
モジュールを組み立てた場合、組み立てた順序（コンフィグレーション）にしたがって入出力のビットが割り当てられてビット構成が決定されます。
一部の電源モジュールや終端モジュールは内部ビット幅を持っておらず、ノードを組み立てた場合のビット構成には影響しません。また、カウンターモジュールやインターフェイスモジュールは入力および出力の両方のビット幅を持っています。これらのモジュールの出力ビット中には制御バイトと呼ばれるコントロールバイトがあり、これを書き換えることによって動作をコントロールします。制御バイトと動作の関係については各々のモジュールの説明を参照してください。

4.2　ビット割当優先順位

表４－２－１はモジュールの一覧およびﾋﾞｯﾄ割当ての優先順位を示します。

＜表４－２－１　モジュール一覧とビット割当て優先順位＞

種別	モデル名	機　　能	入力 ﾋﾞｯﾄ幅	出力 ﾋﾞｯﾄ幅	ﾋﾞｯﾄ割当 優先順位
ﾃﾞｨｼﾞﾀﾙ 入力	750-400	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms	2		B
	750-401	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 0.2ms	2		B
	750-402	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms	4		B
	750-403	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 0.2ms	4		B
	750-405	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　AC230V　応答 10ms	2		B
	750-406	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　AC120V　応答 10ms	2		B
	750-408	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms　負方向	4		B
	750-409	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 0.2ms 負方向	4		B
	750-410	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms　ｾﾝｻｰ対応	2		B
	750-411	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 0.2ms　ｾﾝｻｰ対応	2		B
	750-412	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC48V　応答 3ms	2		B
	750-413	2 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC48V　応答 0.2ms	2		B
	750-414	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC5V　応答 0.2ms	4		B
	750-415	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　AC/DC24V	4		B
	750-430	8 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms	8		B
	750-431	8 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 0.2ms	8		B
	750-432	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms　2 線式	4		B
	750-433	4 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 0.2ms　2 線式	4		B
	750-436	8 ch　ﾃﾞｨｼﾞﾀﾙ入力　DC24V　応答 3ms　負方向	8		B
ﾃﾞｨｼﾞﾀﾙ 出力	750-501	2 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A		2	B
	750-502	2 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　２A		2	B
	750-504	4 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A		4	B
	750-506	2 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A　断線検知機能付	4	4	B
	750-509	2 ch　ﾃﾞｨｼﾞﾀﾙ出力　ｿﾘｯﾄﾞｽﾃｰﾄﾘﾚｰ　AC230、300mA		2	B
	750-516	4 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A　負方向		4	B
	750-519	4 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC5V　20mA		4	B
	750-530	8ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A		8	B
	750-531	4 ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A　2 線式		4	B
	750-536	8ch　ﾃﾞｨｼﾞﾀﾙ出力　DC24V　0.5A　負方向		8	B
リレー 出力	750-512	2 ch　ﾘﾚｰ出力　最大 AC250V2A、+側ｺﾓﾝ、 a 接点		2	B

	750-513	2 ch　ﾘﾚｰ出力　最大 AC250V2A､ﾌﾛｰﾄ型、 a 接点		2	B
	750-514	2 ch　ﾘﾚｰ出力 最大 AC125V　0.5A、ﾌﾛｰﾄ型, c 接点		2	B
	750-517	2 ch　ﾘﾚｰ出力 最大 AC250V　1A、ﾌﾛｰﾄ型, c 接点		2	B
ｱﾅﾛｸﾞ入力	750-452	2 ch　ｱﾅﾛｸﾞ電流入力、0～20mA、外部電源利用型	16×2		A
	750-453	4 ch　ｱﾅﾛｸﾞ電流入力、0～20mA、外部電源利用型	16×4		A
	750-454	2 ch　ｱﾅﾛｸﾞ電流入力、4～20mA、外部電源利用型	16×2		A
	750-455	4 ch　ｱﾅﾛｸﾞ電流入力、4～20mA、外部電源利用型	16×4		A
	750-456	2 ch　ｱﾅﾛｸﾞ電圧入力、±10V、差動入力型	16×2		A
	750-457	4 ch　ｱﾅﾛｸﾞ電圧入力、±10V、外部電源利用型	16×4		A
	750-459	4 ch　ｱﾅﾛｸﾞ電圧入力、0～10V、外部電源利用型	16×4		A
	750-460	4 ch　RTD ｾﾝｻｰ入力	16×4		A
	750-461	2 ch　RTD ｾﾝｻｰ入力	16×2		A
	750-462	2 ch　熱電対入力	16×2		A
	750-465	2 ch　ｱﾅﾛｸﾞ電流入力　0～20mA、ｼｽﾃﾑ電源供給型	16×2		A
	750-466	2 ch　ｱﾅﾛｸﾞ電流入力　4～20mA、ｼｽﾃﾑ電源供給型	16×2		A
	750-467	2 ch　ｱﾅﾛｸﾞ電圧入力、0～+10V、ｼﾝｸﾞﾙｴﾝﾄﾞ入力	16×2		A
	750-468	4 ch　ｱﾅﾛｸﾞ電圧入力、0～+10V、ｼﾝｸﾞﾙｴﾝﾄﾞ入力	16×4		A
	750-469	2 ch　熱電対入力　K ﾀｲﾌﾟ 、断線検知機能付	16×2		A
	750-472	2 ch　ｱﾅﾛｸﾞ電流入力　0～20mA、(15bit)	16×2		A
	750-474	2 ch　ｱﾅﾛｸﾞ電流入力　4～20mA、(15bit)	16×2		A
	750-475	2 ch　ｱﾅﾛｸﾞ電流入力　AC/DC 0～1A、差動入力型	16×2		A
	750-476	2 ch　ｱﾅﾛｸﾞ電圧入力、±10V、(15bit)	16×2		A
	750-477	2 ch　ｱﾅﾛｸﾞ電圧入力、AC/DC 0～+10V、差動入力型	16×2		A
	750-478	2 ch　ｱﾅﾛｸﾞ電圧入力、0～+10V、(15bit)	16×2		A
	750-479	2 ch　ｱﾅﾛｸﾞ電圧入力、±10V、差動入力型（13bit）	16×2		A
	750-480	2 ch　ｱﾅﾛｸﾞ電流入力　0～20mA、差動入力型（13bit）	16×2		A
	750-483	2 ch　ｱﾅﾛｸﾞ電圧入力、0～30V、差動入力型	16×2		A
	750-492	2 ch　ｱﾅﾛｸﾞ電流入力　4～20mA、絶縁差動入力型	16×2		A
ｱﾅﾛｸﾞ出力	750-550	2 ch　ｱﾅﾛｸﾞ電圧出力　0～+10V		16×2	A
	750-552	2 ch　ｱﾅﾛｸﾞ電流出力　0～20mA		16×2	A
	750-554	2 ch　ｱﾅﾛｸﾞ電流出力　4～20mA		16×2	A
	750-556	2 ch　ｱﾅﾛｸﾞ電圧出力　–10V～+10V		16×2	A
	750-557	4 ch　ｱﾅﾛｸﾞ電圧出力　±10V		16×4	A
	750-559	4 ch　ｱﾅﾛｸﾞ電圧出力　–10V～+10V		16×4	A
ｶｳﾝﾀｰ	750-404	1 ch　ｱｯﾌﾟﾀﾞｳﾝｶｳﾝﾀｰ　DC24V,　100kHz	48	48	A
	750-638	2ch　ｱｯﾌﾟﾀﾞｳﾝｶｳﾝﾀｰ　DC24V　500Hz	48	48	A
ｼﾘｱﾙｲﾝﾀﾌｪｰｽﾓｼﾞｭｰﾙ	750-650	RS232C ｲﾝﾀﾌｪｰｽﾓｼﾞｭｰﾙ、ﾊﾟﾘﾃｨ無、8 ﾋﾞｯﾄ、ｽﾄｯﾌﾟﾋﾞｯﾄ 1	24	24	A
	750-650 /003-000	RS232C ｲﾝﾀﾌｪｰｽﾓｼﾞｭｰﾙ、通信条件可変タイプ	24～40	24～40	A
	750-653	RS485 ｲﾝﾀﾌｪｰｽﾓｼﾞｭｰﾙ、ﾊﾟﾘﾃｨ無、8 ﾋﾞｯﾄ、ｽﾄｯﾌﾟﾋﾞｯﾄ 1	24	24	A
	750-653 /003-000	RS485 ｲﾝﾀﾌｪｰｽﾓｼﾞｭｰﾙ、通信条件可変タイプ	24～40	24～40	A
	750-601	電源ﾓｼﾞｭｰﾙ　ﾋｭｰｽﾞ付　DC24V　6.3A			
	750-602	電源ﾓｼﾞｭｰﾙ　DC24V			
	750-609	電源ﾓｼﾞｭｰﾙ　ﾋｭｰｽﾞ付　AC230V　6.3A			
	750-610	電源ﾓｼﾞｭｰﾙ　ﾋｭｰｽﾞ付　DC24V　6.3A 診断機能付	2		B
	750-611	電源ﾓｼﾞｭｰﾙ　ﾋｭｰｽﾞ付　AC230V　6.3A　診断機能付	2		B
	750-612	電源ﾓｼﾞｭｰﾙ　電源ﾓｼﾞｭｰﾙ　AC/DC　230V,　ﾋｭｰｽﾞなし			
	750-613	電源ﾓｼﾞｭｰﾙ　システムおよびフィールド用 DC24V			
	750-614	電源端子拡張ﾓｼﾞｭｰﾙ			
	750-615	電源ﾓｼﾞｭｰﾙ　ﾋｭｰｽﾞ付 AC120V　6.3A			
	750-600	終端ﾓｼﾞｭｰﾙ　＊			

＊　750-600　終端モジュールは必ず最後に設置してください。

ビット割当ては以下のような簡単なルールにしたがって割り当てられます。同じモジュール構成であっても、設置順序が異なると、ビット割当て（フィールドバスを経由して外部から入出力したときのビット構成順序）が異なります。

＜ビット割り当てルール＞

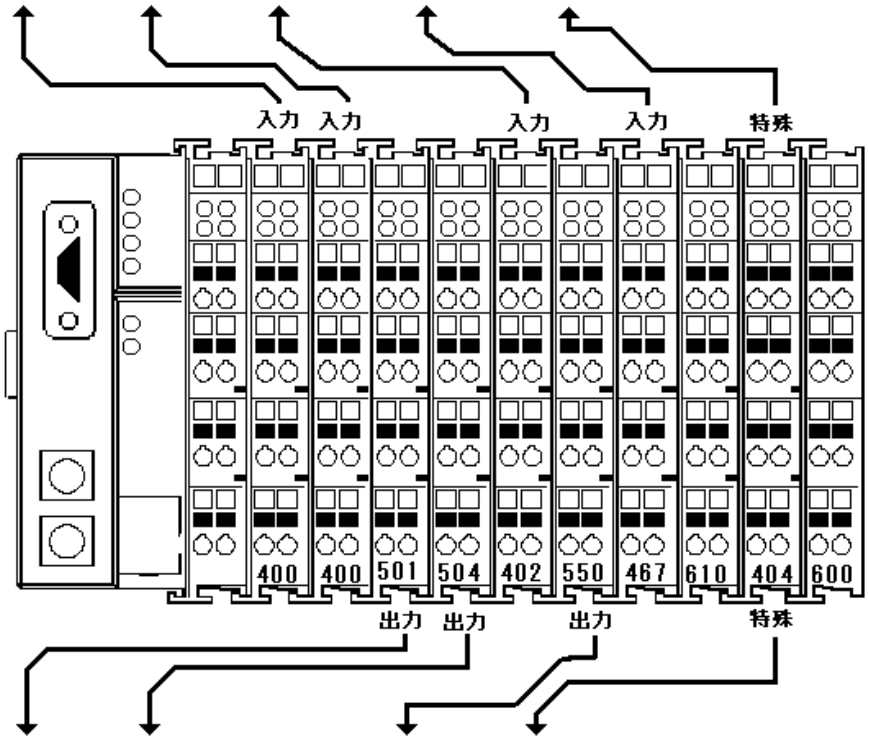
① 入力と出力のビット割当ては別々に行われます。
② バスカップラーに近いモジュールから順番にビットが割り当てられます。
③ ディジタル入出力モジュールとアナログモジュール等が混在している場合、アナログモジュールや特殊モジュール（優先順位が A で表されるモジュールがディジタル入出力より先にビットが割り当てられます。
④ アップダウンカウンター、各種インターフェイスモジュールは入力と出力の両方にビット割当てされます。
⑤ 内部ビット幅を持たないモジュールはどの位置に挿入してもビット構成には影響しません。

4.3　ビット割当の例

図４－３－１はモジュールのコンフィグレーションとビット割当ての一例を示しています。入力においても出力においてもアナログモジュールや特殊モジュールが優先されて先にビットが割り当てられ、その後ディジタルモジュールのビットが割り当てられます。ビットのアドレスは４桁の１６進数で示されており、これはフィールドバスを経由して読み書きするときのアドレスになります。
標準添付の開発支援 DLL を使ってソフト開発を行う場合はこれらのルールを気にすること無くプログラム開発ができます。また、アナログモジュールに関するビットデータと物理量の変換も自動的に行われますので、ビット割当てルールやアドレスを気にすることなくソフト開発に専念できます。開発支援 DLL につきましては、「第６章　開発支援ソフト（Dynamic　Link　Library）」を参照してください。

図４－３－１

入力	ﾓｼﾞｭｰﾙ	750-467	750-404	750-400	750-400	750-402	750-610
	内部ﾋﾞｯﾄ幅	16×２	４８	２	２	４	２
	ｱﾄﾞﾚｽ	0000～	0020～	0050～	0052～	0054～	0058～



･出力	ﾓｼﾞｭｰﾙ	750-550	750-404	750-501	750-504
	内部ﾋﾞｯﾄ幅	16×２	４８	2	4
	割当ｱﾄﾞﾚｽ	0000～	0020～	0050～	0052～

5　通信プロトコル

5.1　シリアル通信 Modbus プロトコル

5.2　データ読み出しと書き込みの機能分類

スレーブであるワゴ I/O からデータを読み出す、あるいはデータを書き込むという動作は、マスターから出力されるリクエスト信号に含まれる機能コードによって区別します。同一のノード内に入力モジュールと出力モジュールが混在していても、読み出し動作（入力モジュール）と書き込み動作（出力モジュール）はそれぞれ独立した機能として扱います。すなわち**読み出し動作時は入力モジュールのみが対象**となり、**書き込み動作時は出力モジュールのみが対象**となります。

読み出しおよび書き込み動作は、以下の表のように**７種類の機能**に分類されます。５－１－２以降に各々の動作の詳細を説明してありますが、表中の「**機能コード」は Modbus の機能コードに対応**しています。

●ワゴ I/O からデータを読み出す（入力）

分類	機能	対象モジュール	機能コード
①	ビット単位で範囲を指定してデータを読みだす。	ディジタル入力モジュール	「０１」または「０２」
②	ワード単位で範囲を指定してデータを読み出す	アナログ入力モジュールおよびディジタル入力モジュール	「０３」または「０４」
③	電源投入以降に実行されたコマンドの回数を読み出す	バスカップラ	「0B」

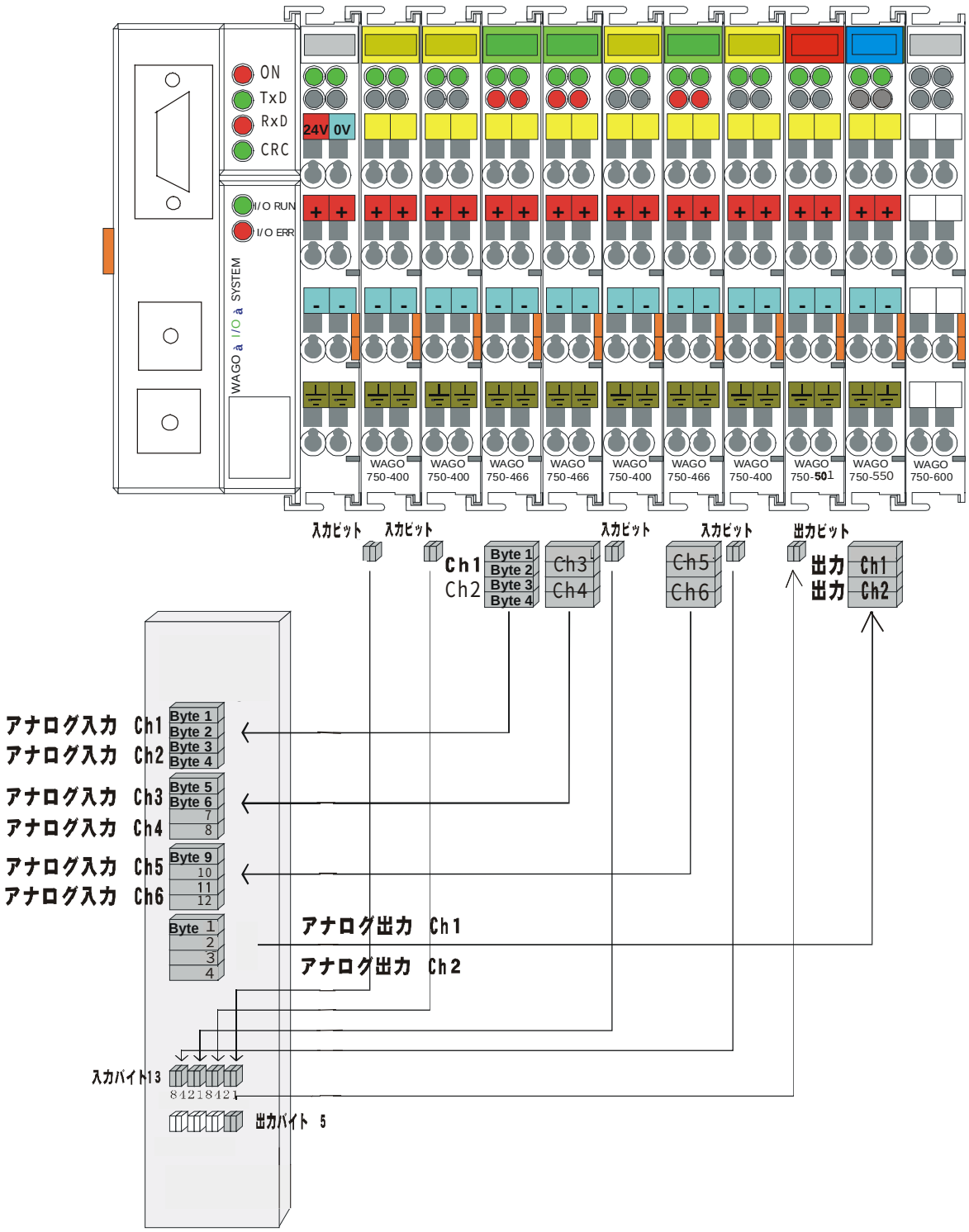
●ワゴ I/O へデータ書き込む（出力）

分類	機能	対象モジュール	機能コード
④	単一のビットを指定してデータを書き込む	ディジタル出力モジュール	「０５」
⑤	範囲を指定して複数ビットにデータを書き込む	ディジタル出力モジュール	「０Ｆ」
⑥	単一のワードを指定してデータを書き込む	アナログ出力モジュールおよびディジタル出力モジュール	「０６」
⑦	範囲を指定して複数ワード単位でデータを書き込む	アナログ出力モジュールおよびディジタル出力モジュール	「１０」

尚、この章に記載されている内容は Modbus の通信仕様について説明したもので、標準添付の開発支援ＤＬＬを使用してソフト開発する場合は考慮する必要はありません

図５－１－１

●入力、出力モジュールは混在していても独立した機能として扱う



5.2.1　　読み出し、書き込みコマンドの送出方法

マスター（ホストコンピューター）とスレーブ（ワゴＩ／Ｏ）は**１対１の通信を基本とします。**マスター側からのリクエスト信号を受け取ったワゴ I/O はそのリクエスト信号が自分のノード（アドレス）に対するリクエストであるかどうかを判断し、自分のノードに対するリクエストであれば、必要な情報を含んだアンサーバック信号を返します。

●リクエスト信号（マスター　→　ワゴ I/O）の例

	先頭	アドレス	機能コード	開始アドレス	範囲（ビット）	エラーチェック	フレームエンド
アスキー文字	：	０Ｂ	０１	０００Ｏ	０００８	ＥＣ	
アスキーコード	3A	3042	3031	30303030	30303038	4543	DA
バイナリコード	３フレーム分の間隔	０Ｂ	０１	０００Ｏ	０００８	３Ｄ６６	３フレーム分の間隔

アスキー文字の場合先頭は「：」（コロン）で始まります。

スレーブのアドレスを指定します。

動作の内容を機能コードで指定します。

数え始めのスタートアドレスです。この場合はバスカップラに一番近いディジタル入力ビットから数え始めます。（１６進数で表現します。）

数えはじめから何ビット分を範囲にするかを決めます。１６進数で表わします。ワゴ I/O 側では８ビット（バイト）単位で処理しますので、例えば「０００Ｄ」（１３）と設定した場合でもワゴ I/O 側は１６ビット（２バイト）分をデータ領域として確保し、範囲に指定されていない３ビット 分はデータが無いものとしてアンサーバック信号に反映します。０００Ｏ～００FF までが設定範囲です。

送信するデータのＣＲＣ／ＬＲＣコードが添付されます。DIP ＳＷ設定でこれを無視するときは必要ありません。

●　アンサーバック信号：（　ワゴ I/O　→　マスター）の例

	先頭	アドレス	機能コード	バイトカウント	データ	エラーチェック	フレームエンド
アスキー文字	：	０Ｂ	０１	０１	１２	Ｅ１	
アスキーコード	3A	3042	3031	3031	3132	4531	DA
バイナリコード	３フレーム分の間隔	０Ｂ	０１	０１	１２	Ｄ２５Ｄ	３フレーム分の間隔

アンサーバック信号も先頭は「：」で始まります。

応答したスレーブが自分のアドレスを確認の意味で送信します。

リクエスト信号で要求された機能コードを確認のため返信します。

指定された範囲が１バイト分以下であることを示します。リクエスト信号で指定されたビット範囲が１バイト単位で区切れない場合は範囲に満たないビットはデータなし（0）として送信します。

ワゴ I/O から送信されるデータにもＣＲＣ／ＬＲＣエラーチェックコードが添付されます。

データは４ビット分×２セット（１バイト）ごとに２桁の１６進数を用いて表わします。この場合は上位の４ビットが「1」、下位の４ビットが「2」ですから、ビット構成に直すと以下のような並びになります。

2　　1

○ ● ○ ○　　● ○ ○ ○

(1) (2) (4) (8)　(1) (2) (4) (8)　　(●：入力　オン)

←　バスカップラに近い方向　　(○：入力　オフ)

5.2.2　　ワゴ I/O からデータを読み出す

①ビット単位で範囲を指定してデータを読み出す（機能コード　「０１」または「０２」）

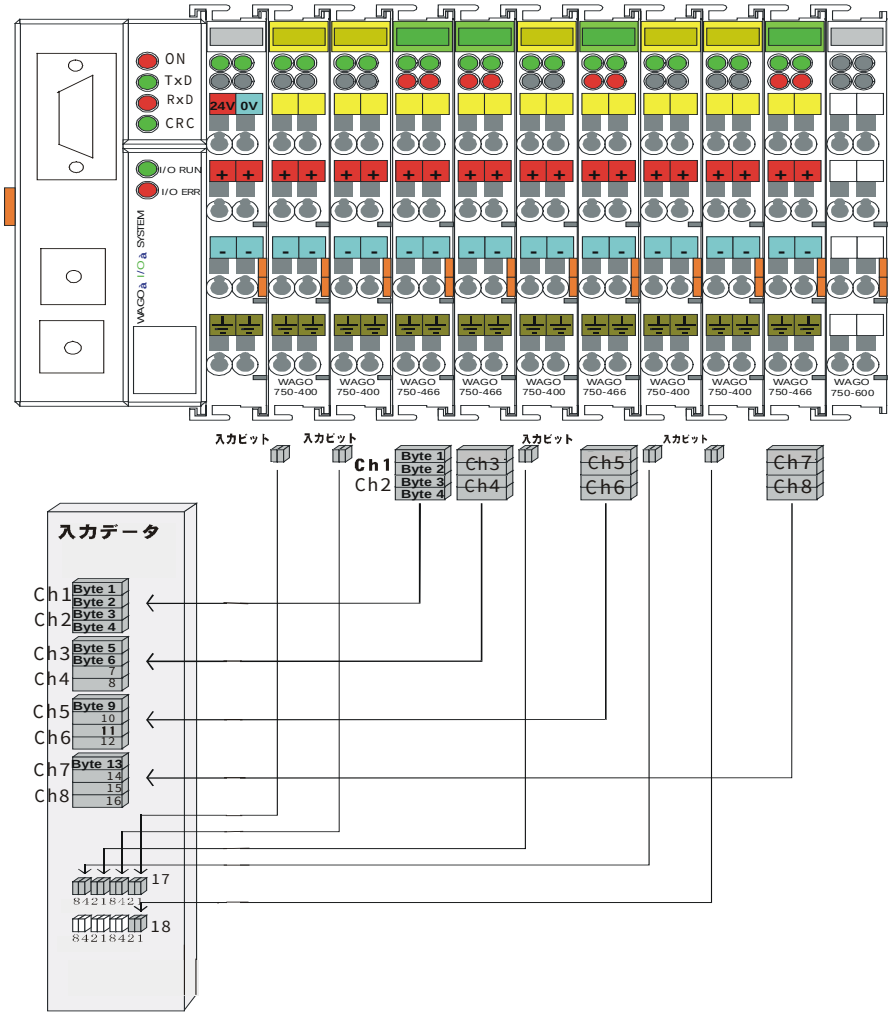
このコマンドはビット単位で開始位置を設定、ビット単位で範囲を指定して、その入力値（「１」か「０」また　は「High」か「Low」）を読み出します。リクエスト信号とアンサーバックの信号の例は５－１－２で説明したとおりです。このコマンドはディジタル入力モジュールのみが対象となります。また、ビット単位での最高点数は２５６点（３２バイト）分です。すなわちリクエスト信号の中の開始アドレスにおける最高値は４桁１６進数で「００FF」です。
アンサーバック信号中のデータは８ビット単位（バイト単位）で区切られ、バスカップラに近いディジタル入力モジュールの４ビット分の１６進数値がデータの下位桁によって表され、バスカップラから遠いモジュールの４ビット分の１６進数値が上位桁に表されます。

②範囲を指定してワード単位でデータを読み出す（機能コード　「０３」または「０４」）

このコマンドは１６ビット単位（ワード単位）で範囲を指定してワゴＩ／Ｏからデータを読み出します。アンサーバック信号中の１６進数値の構成は４ビット×２セット（１バイト）を単位として上位と下位２桁ずつの組み合わせで表します。またアナログモジュールが挿入されていた場合はアナログ入力モジュール（１チャンネルあたり１６ビット）が優先されます。すなわちディジタル入力モジュールのデータはアナログデータの後に続くことになります。読み出せる最大の数はバイトカウンタ２桁が限界になりますので、バイトカウントで FF（２５６）、１６ビットレジスター（２バイト）の数に直すと１２８が最大になります。

図５－１－３

●ワゴI/0からデータを読み出す（入力）



●リクエスト信号（マスター　→　ワゴ I/O）

	先頭	アドレス	機能コード	開始アドレス	範囲（ワード）	エラーチェック	フレームエンド
アスキー文字	:	0B	03	0000	0002	F0	
アスキーコード	3A	3042	3033	30303030	30303032	4630	DA
バイナリコード	３フレーム分の間隔	0B	03	0000	0002	C4A1	３フレーム分の間隔

数え始めのスタートアドレスをワード単位で指定します。１６進数で設定します。例えば、２チャンネルのアナログ入力モジュール１個が挿入されているときには、これが優先されますので、この場合は１ワード目（００００）がアナログCH1, ２ワード目（０００１）がアナログCH2, その後にディジタル入力ビットが、バスカップラに近い方から割り当てられます。

アナログモジュールのデータを無視して３ワード目からディジタル入力データだけを読みたい場合はここを「０００２」と設定します。

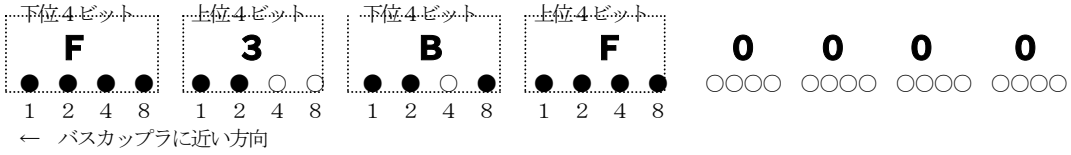
スタートアドレスビットから何ワード分を範囲にするかをせ指定します。１６進数で単位はワードです。この場合は２ワード分、すなわち３２ビットの範囲を設定しています。

● アンサーバック信号：(ワゴ I/O → マスター)

	先頭	アドレス	機能コード	バイトカウント	データ下位レジスタ０	データ上位レジスタ１	エラーチェック	フレームエンド
アスキー文字	:	0B	03	04	3FFB	0000	B4	
アスキーコード	3A	3042	3033	3034	33464642	30303030	4234	DA
バイナリコード	３フレーム分の間隔	0B	03	04	3FFB	0000	B4	３フレーム分の間隔

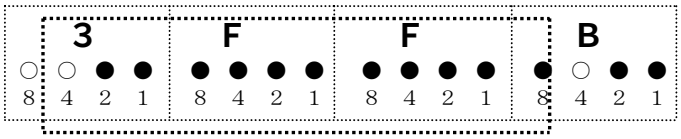
指定された範囲が、バイト単位に直すと４バイト分であることを意味します。

データは４ビット分×２セット（１バイト）ごとに４桁の１６進数（１ワード）を用いて表わします。この場合は下位のワード分が「**3FFB**」で、この１６ビット分が下位ワード（レジスター０）になります。これを分解すると以下のようなビット構成になります。仮にすべてがディジタル入力である場合はビットの並びはバスカップラに近い方から以下の並びと同じになります。



(●：入力 オン)　(○：入力 オフ)

２CHアナログ入力モジュールが挿入されているときは、１チャンネルあたり１６ビットが割り当てられます。下位ワード（レジスタ０）の１２ビット分を使います。「３FFB」の場合は、以下のように分解され、１２ビットのうちの１１ビット分がオンの状態を示します。ただし、通常は位３ビットは使用せず、ビットがＨｉのままになりますので数値は**「３FFF」になります。**



A／D変換時の有効な１２ビット

仮に０～１０V のアナログ入力モジュールに適用すると以下のようになります。

$$\frac{2^{11}\times 0+2^{10}\times 1+2^{9}\times 1+2^{8}\times 1+2^{7}\times 1+2^{6}\times 1+2^{5}\times 1+2^{4}\times 1+2^{3}\times 1+2^{2}\times 1+2^{1}\times 1+2^{0}\times 1}{2^{12}}\times 10\mathrm{V}$$

$$= \frac{2047}{4095}\times 10\mathrm{V} = 5\mathrm{V}$$

③実行されたコマンド回数を読み出す（機能コード「０Ｂ」）

このコマンドはバスカップラ内にあるカウンタのカウント数値を読み出します。バスカップラ内のコマンドカウンタは電源が投入された後、正常に動作が行われたコマンドの回数をカウントしていますので、マスター側から送り出したコマンドの数とこの数をチェックすることによってそのノードとの通信が正常に行われているかどうかを確認することができます。

●リクエスト信号（マスター → ワゴ I/O）

	先頭	アドレス	機能コード	エラーチェック	フレームエンド
アスキー文字	:	0B	0B	EA	
アスキーコード	3A	3042	3042	4541	DA
バイナリコード	３フレーム分の間隔	0B	0B	4747	３フレーム分の間隔

●アンサーバック信号（ワゴ I/O　→　マスター）

	先頭	アドレス	機能コード	ステータスバイト	イベント回数	エラーチェック	フレーム エンド
アスキー文字	：	0B	0B	0000	0003	E7	
アスキーコード	3A	3042	3042	30303030	30303033	4537	DA
バイナリコード	3フレーム分の間隔	0B	0B	0000	0003	E4A0	3フレーム分の間隔

回数は4桁の１６進数で表されます。この場合は電源投入時から現在まで、このスレーブに対して正常なコマンドのやり取りが行われた回数が3回であることを示しています。

このステータスバイトはワゴ I/0 では使いませんので常に「００００」となります。

5.2.3　　ワゴ I/O へデータを書き込む

④単一のビットを指定してデータを書き込む（機能コード「０５」）

このコマンドは単一のビットを指定して、ＯＮ／ＯＦＦの出力を実行します。出力の実行は単一のビットですが、設定できる範囲は０００００～００ＦＦまでの２５６ポイントの範囲です。この機能の対象モジュールはディジタル出力モジュールのみです。アナログ出力モジュールは関係ありません。

●リクエスト信号（マスター　→　ワゴ I/O）

	先頭	アドレス	機能コード	ビットアドレス	命令コード	エラーチェック	フレームエンド
アスキー文字	：	0B	05	0000	FF00	F1	
アスキーコード	3A	3042	3035	30303030	46463030	4631	DA
バイナリコード	3フレーム分の間隔	0B	05	0000	FF00	8C90	3フレーム分の間隔

操作をするビットのアドレスです。ディジタル出力モジュールのみが対象となります。０００００～００ＦＦまでの２５６ポイントの範囲で指定します。

指定したビットについて、ONにするかＯＦＦにするかを設定します。ＯＦＦの状態からONにするときには「**ＦＦ００**」とし、ONの状態からＯＦＦの状態に戻すには「**０００００**」と設定します。

●アンサーバック信号：（　ワゴ I/O　→　マスター）

	先頭	アドレス	機能コード	ビットアドレス	命令コード	エラーチェック	フレームエンド
アスキー文字	：	0B	05	0000	FF00	F1	
アスキーコード	3A	3042	3035	30303030	46463030	4631	DA
バイナリコード	3フレーム分の間隔	0B	05	0000	FF00	8C90	3フレーム分の間隔

ワゴ I/O（スレーブ）はデータ書き込みに成功した場合には同じデータをアンサーバックとして返信します。

⑤ 範囲を指定して複数ビットにデータを書き込む（機能コード「０F」）

このコマンドは複数のビットを指定して、データを書き込みます。アナログモジュールは関係ありません。

●リクエスト信号（マスター　→　ワゴ I/O）

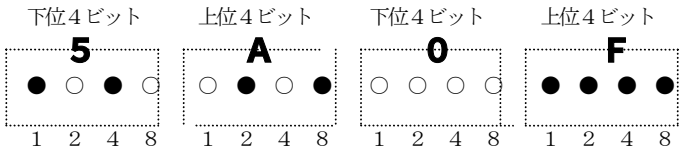
	先頭	アドレス	機能コード	開始アドレス	範囲(ビット)	バイトカウンター	出力データ	エラーチェック	フレームエンド
アスキー文字	:	0B	0F	0000	0010	02	A5F0	3F	
アスキーコード	3A	3042	3046	30303030	30303130	3032	41354630	3346	DA
バイナリコード	３フレーム分の間隔	0B	0F	0000	0010	02	A5F0	E794	３フレーム分の間隔

数え始めのスタートアドレスをビット単位で指定します。１６進数で設定します。ディジタル入力ビットが、バスカップラに近い方から割り当てられます。スタートアドレスは０００００からスタートして１６進数で数えていきますので、１０進で１６番目が「０００F」、１７番目が「００１０」となります。アナログモジュールは無視されます。

スタートアドレスビットから何バイト分を範囲にするかを指定します。この場合は１６ビット分を範囲設定しているので２バイト分を設定しています。出力範囲ビットカバーする数値に設定してください。

出力する範囲をビット単位で設定します。この場合はスタートアドレス「００００」から数えて「００１０」（１６ビット分）を範囲とするという意味になります。

データは４ビット分×２セット（１バイト）ごとに４桁の１６進数（１ワード）を用いて表わします。この場合は指定された２バイト分（１ワード分）が「A５F０」で、これを分解すると以下のようなビット構成になります。ビットの並びはバスカップラに近い方から以下の並びと同じになります。



← バスカップラに近い方向

(●：入力　オン)　　(○：入力　オフ)

●アンサーバック信号：(　ワゴ I/O　→　マスター)

	先頭	アドレス	機能コード	開始アドレス	範囲（ビット）	エラーチェック	フレームエンド
アスキー文字	:	0B	0F	0000	0010	D6	
アスキーコード	3A	3042	3046	30303030	30303130	4436	DA
バイナリコード	３フレーム分の間隔	0B	0F	0000	0010	54AD	３フレーム分の間隔

⑥ 単一のワードを指定してデータを書き込む

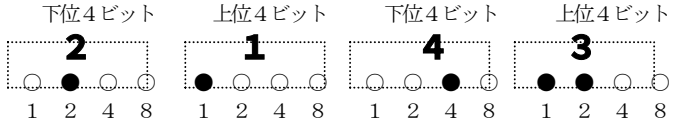
このコマンドは単一のレジスタ（１６ビット分）に対してデータを書き込むためのコマンドです。例えばアナログ出力モジュールが挿入されている場合にはこれが優先されます。アドレス「０００００」（レジスタ０）がアナログ１CH 目、アドレス「０００１」（レジスタ１）がアナログ２CH 目になります。

●リクエスト信号（マスター　→　ワゴ I/O）

	先頭	アドレス	機能コード	開始アドレス	書き込みデータ(単一レジスタへ)	エラーチェック	フレームエンド
アスキー文字	:	0B	06	0000	1234	A9	
アスキーコード	3A	3042	3036	30303030	31323334	4139	DA
バイナリコード	３フレーム分の間隔	0B	06	0000	1234	8C17	３フレーム分の間隔

数え始めのスタートアドレスをワード単位で指定します。１６進数で設定します。例えば、２チャンネルのアナログ出力モジュール１個が挿入されているときには、これが優先されますので、この場合は１ワード目（００００）がアナログ CH1，２ワード目（０００１）がアナログ CH2，その後にディジタル出力ビットが、バスカップラに近い方から割り当てられます。
アナログ出力モジュールのデータを無視して３ワード目からディジタル出力モジュールにのみデータを書き込みたい場合はここを「０００２」と設定します。

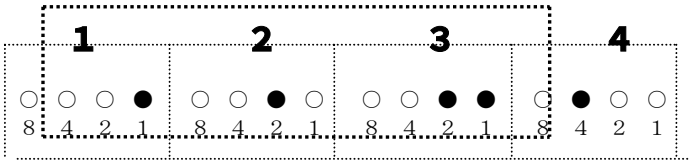
データは４ビット分×２セット（１バイト）ごとに４桁の１６進数（１ワード）を用いて表わしますこのコマンドはレジスタ１個分、すなわち１６ビット分の書き込みしかできませんので、４ビット×２セット×２セット分のデータを以下のように４桁の１６進数で表現します。仮にすべてがディジタル出力モジュールである場合はビットの並びはバスカップラに近い方から以下の並びと同じになります。



← バスカップラに近い方向

(●：入力　オン)　　(○：入力　オフ)

２CHアナログ出力モジュールが挿入されているときは、アドレスを「０００００」に設定してデータを送ると、まずアナログ出力の CH1 にデータが適用されます。アナログ出力１CH あたり１６ビット（１ワード）が割り当てられ、以下のように１２ビット分を使います。例えば１０V の出力モジュールで「１２３４」を出力すると１．４２V になります。



A／D変換時の有効な１２ビット

$$\frac{2^{11}\times 0+2^{10}\times 0+2^{9}\times 1+2^{8}\times 0+2^{7}\times 0+2^{6}\times 1+2^{5}\times 0+2^{4}\times 0+2^{3}\times 0+2^{2}\times 1+2^{1}\times 1+2^{0}\times 0}{2^{12}}\times 10V$$

$$= \frac{582}{4095} \times 10V = 1.42V$$

● アンサーバック信号：（ ワゴ I/O　→　マスター）

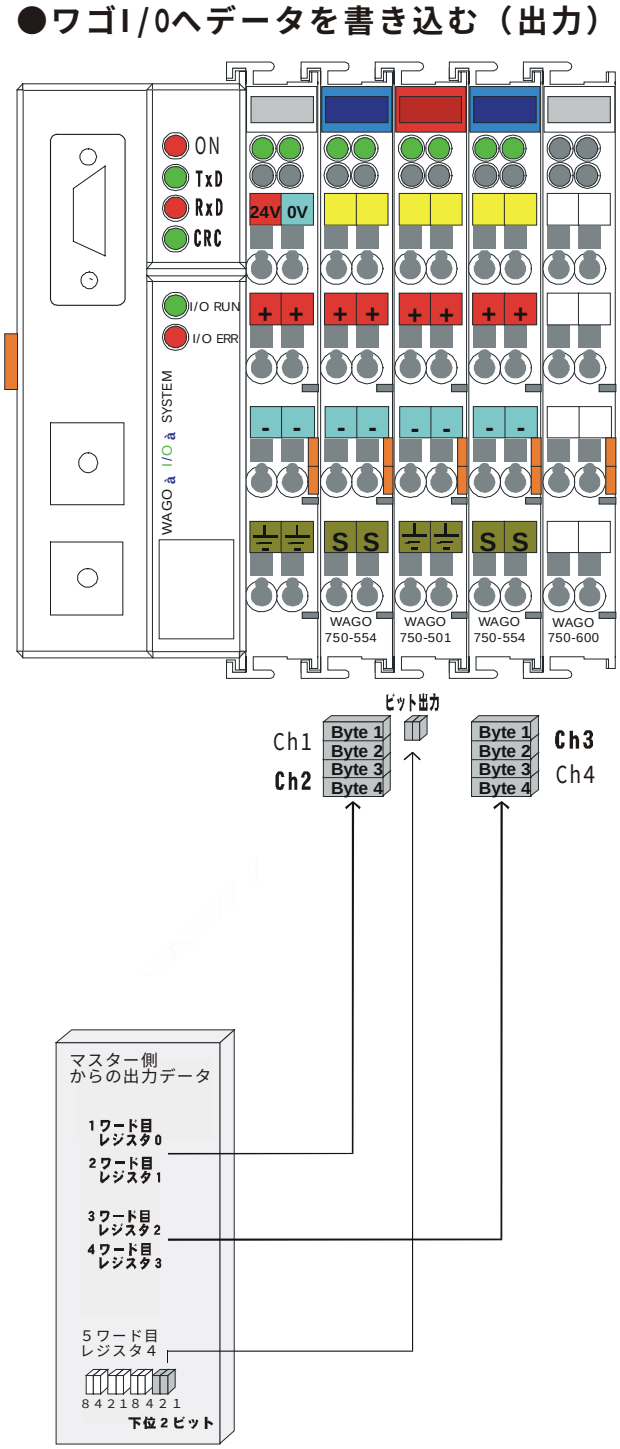
	先頭	アドレス	機能コード	データ下位レジスタ０	データ上位レジスタ１	エラーチェック	フレームエンド
アスキー文字	:	0B	06	0000	1234	A9	
アスキーコード	3A	3042	3036	30303030	31323334	4139	DA
バイナリコード	３フレーム分の間隔	0B	06	0000	1234	8C17	３フレーム分の間隔

アンサーバック信号は確認の意味でリクエスト信号と同じ内容を返信します。

⑦ 範囲を指定して複数ワード単位でデータを書き込む

このコマンドは複数のレジスタ（１６ビット分×N）に対してデータを書き込むためのコマンドです。アナログ出力モジュールが挿入されている場合にはこれが優先されて割り当てられます。出力データはレジスタの低い方から順番に４桁の１６進数で送出します。

図５－１－４



●リクエスト信号（マスター　→　ワゴ I/O）

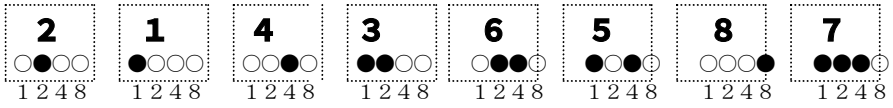
	先頭	アドレス	機能コード	開始アドレス（ワード単位）	範囲（ワード単位）	バイトカウンター	出力データレジスタ０	出力データレジスタ１	エラーチェック	フレームエンド
アスキー文字	：	０Ｂ	１０	００００	０００２	０４	１２３４	５６７８	ＣＢ	
アスキーコード	３Ａ	３０４２	３１３０	３０３０３０３０	３０３０３０３２	３０３４	３１３２３３３４	３５３６３７３８	４３４２	ＤＡ
バイナリコード	３フレーム分の間隔	０Ｂ	１０	００００	０００２	０４	１２３４	５６７８	Ａ９４３	３フレーム分の間隔

数え始めのスタートアドレスをワード単位で指定します。１６進数で設定します。例えば、２チャンネルのアナログ出力モジュール１個が挿入されているときには、これが優先されますので、この場合は１ワード目（００００）がアナログ CH1, ２ワード目（０００１）がアナログ CH2, その後にディジタル出力ビットが、バスカップラに近い方から割り当てられます。
　アナログ出力モジュールのデータを無視して３ワード目からディジタル出力モジュールにのみデータを書き込みたい場合はここを「０００２」と設定します。

出力する範囲をワード単位で設定します。この場合はスタートアドレス「００００」から数えて「０００２」（２ワード分＝３２ビット分）を範囲とするという意味になります。

スタートアドレスから何バイト分を範囲にするかを指定します。この場合は３２ビット分を範囲設定しているので４バイトを設定しています。出力範囲ビットカバーする数値に設定してください。

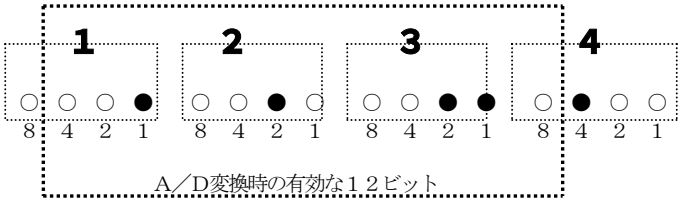
　データは４ビット分×２セット（１バイト）ごとに２桁の１６進数を用いて表わしますこのコマンドはレジスタを複数個分、すなわち１６ビット分×ｎ個の書き込みが可能です。１６ビットごとに４桁の１６進数で置き換えながらデータを作ります。仮にすべてがディジタル出力モジュールである場合はビットの並びはバスカップラに近い方から以下の並びと同じになります。



← バスカップラに近い方向

(●：入力　オン)　　(○：入力　オフ)

　アナログ出力モジュールが挿入されているときは、アドレスを「００００」に設定してデータを送ると、まずアナログ出力の CH1 にデータ
が適用されます。アナログ出力１CH あたり１６ビット（１ワード）が割り当てられ、以下のように１２ビット分を使います。例えば「１２３４」のデータを CH１に送ると、以下のように　V が出力されます。



$$\frac{2^{11}\times 0+2^{10}\times 0+2^{9}\times 1+2^{8}\times 0+2^{7}\times 0+2^{6}\times 1+2^{5}\times 0+2^{4}\times 0+2^{3}\times 0+2^{2}\times 1+2^{1}\times 1+2^{0}\times 0}{2^{12}}\times 10V$$

$$= \frac{582}{4095} \times 10V = 1.42V$$

●アンサーバック信号：(　ワゴ I/O　→　マスター)

	先頭	アドレス	機能コード	開始アドレス「ワード単位」	出力の範囲「ワード単位」	エラーチェック	フレームエンド
アスキー文字	：	０Ｂ	１０	００００	０００２	Ｅ３	
アスキーコード	3A	3042	3130	30303030	30303032	4533	E3
バイナリコード	３フレーム分の間隔	０Ｂ	１０	００００	０００２	４１６２	３フレーム分の間隔

アンサーバック信号は確認のため出力を指定した開始のアドレス（ワード単位）と出力した範囲をワード単位で返信します。

5.2.4 （参考）CRC（巡回符号方式）コードの生成

CRC コード（The Cyclical Redundancy Check）は１６ビットの２進数値、２バイトで構成されます。CRC コードは一方の機器側で計算して生成され、メッセージデータの最後に付加されます。もう一方の機器はメッセージデータを受信しながら CRC を自ら計算し、受信したメッセージデータに添付されてきた CRC との照合を実行します。この２つのデータが一致しない場合は通信上のミスがあることがわかります。このようにして通信エラーを排除するわけです。

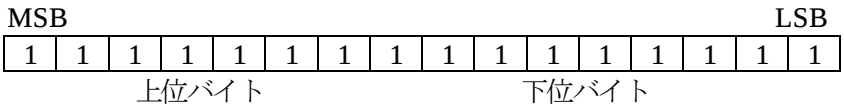
CRC の計算は、最初に１６ビットのレジスタにすべて“1”をロードします。このレジスタに、連続する８ビットのバイト単位で次々と処理して演算することによって計算されます。それぞれの８ビット単位のデータにこの演算は適用されますが、スタートおよびストップビット、パリティビットについては CRC 演算は適用されません。

CRC の演算中は、８ビット単位の文字データがレジスタ内のデータの各ビットと EX-OR 処理されて、その結果が LSB の方へシフトされ、MSB には「0」が入力されます。そして LSB のビットをチェックします。もし、LSB が「1」であれば、レジスタは最初の設定値と EX-OR 処理され、LSB が「0」の場合は何も処理は行われません。

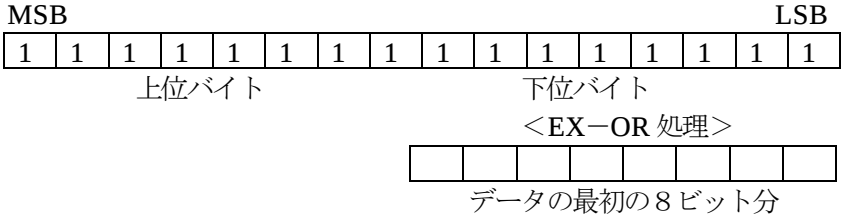
この処理は８ビット分８回シフトして繰り返されます。最後の８回目の処理が行われた後、次の８ビットデータが、シフトレジスタにある現在値と EX-OR 処理され、同じように８回の処理が行われます。このようににしてすべてのデータを処理した結果、最後にシフトレジスタに残る値が CRC データです。

＜具体的な計算処理方法＞

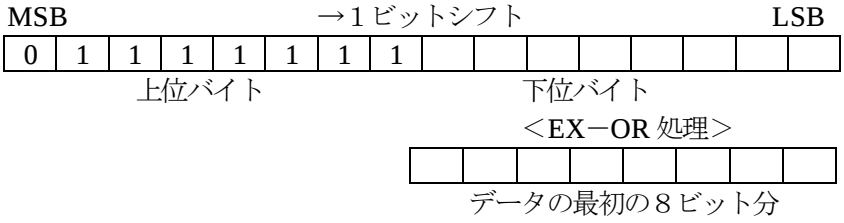
1）まず、１６ﾋﾞｯﾄのシフトレジスタに１６進数で FFFF, すべてに“1”をセットします。
このレジスタを CRC レジスタと呼ぶことにします。



2）データの最初の８ビット分（１バイト分）をこの CRC レジスタの下位バイトと EX-OR 処理してこの結果を CRC レジスタに格納します。



3）このCRCレジスタを１ビット右（LSBの方向）へシフトして、一番左（MSB）には「０」を格納します。



4）このとき、LSBに格納されている値は「０」の場合は３）の動作を繰り返します。LSBの値が「１」の場合はCRCレジスタのデータと多項式データ　A００１HEX（1010 0000 0000 0001）とEX-OR処理を行います。

5）８ビットシフトして３）あるいは４）を８回繰り返します。これが完了した時点で最初の８ビットの処理が完了します。

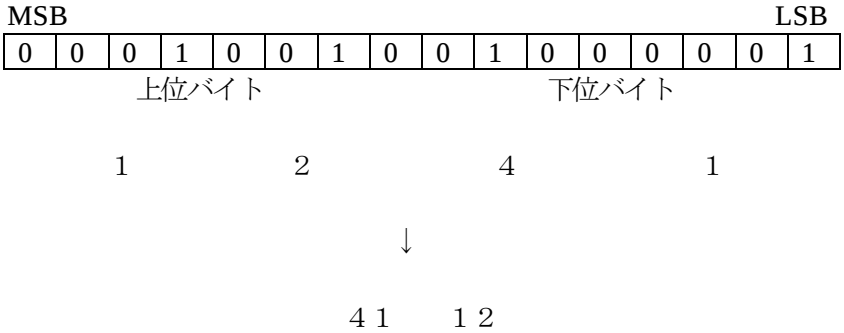
6）２）～５）までの処理を次の８ビットについて行います。

この結果、このCRCレジスタに残った値がCRC値です。

7）このCRC値がデータメッセージに添付されるときは、上位バイトと下位バイトは以下に記述されるようにスワップされます。

【CRCデータをメッセージデータに添付する例】

１６ビットのCRC（２×８ビットデータ）がデータに添付されるとき、先に低いオーダーの８ビットが送信され、次に高いオーダーの８ビットが送信されます。例えばCRC値が１２４１HEX（0001 0010 0100 0001）だとします。この場合、CRCデータはメッセージデータに引き続いて４１（HEX）１２（HEX）と送信されます。



＜C言語でCRCデータを生成するプログラムサンプル＞

以下にC言語でCRCデータを計算する例を示します。CRCデータは２つの数字配列で既にテーブルで用意されており、関数でそれぞれのインデックスを指定するだけで必要な配列のデータを得ることができます。１６ビットデータのうちの上位バイトCRC値は２５６個、下位バイトが２５６個、計算され得るすべてのCRCデータのテーブルになっています。このテーブルを用いると、CRC生成アルゴリズムを用いて計算するよりも速くCRC値を算出することががができるので、コンパイル後独自の装置

などに組み込むことができます。

注意）

このサンプルプログラム中で行われる演算では、上位バイトと下位バイトのスワップは自動的に行われていますので、このプログラムで得られる CRC データはそのまま送信データに添付するだけで OK です。

このプログラム中には２つの引数があります。

unsigned char　*puchMsg ;　メッセージデータバッファへのポインタ
unsigned short us　Datalen ;　メッセージデータのバイトの数

＜以下サンプルプログラム例＞

```
unsigned short CRC16(puchMsg, usDataLen)

unsigned char *puchMsg ;              /* message to calculate CRC
upon */
unsigned short usDataLen ;            /* quantity of bytes in message
*/

{
        unsigned char uchCRCHi = 0xFF ;     /* high CRC byte
initialized */
        unsigned char uchCRCLo = 0xFF ;     /* low CRC byte
initialized   */
        unsigned uIndex ;                           /* will index into CRC
lookup*/
                                                                        /*
table
  */

        while (usDataLen--)                 /* pass through message buffer
  */
                {
                uIndex = uchCRCHi ^ *puchMsgg++ ;/* calculate the CRC
 */
                uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex} ;
                uchCRCLo = auchCRCLo[uIndex] ;
                }

        return (uchCRCHi << 8 | uchCRCLo) ;
}
```

＜上位バイトの CRC 値テーブル＞

/* Table of CRC values for high-order byte */

```
static unsigned char auchCRCHi[] = {
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;
```

下位バイト CRC 値テーブル

/* Table of CRC values for low-order byte */

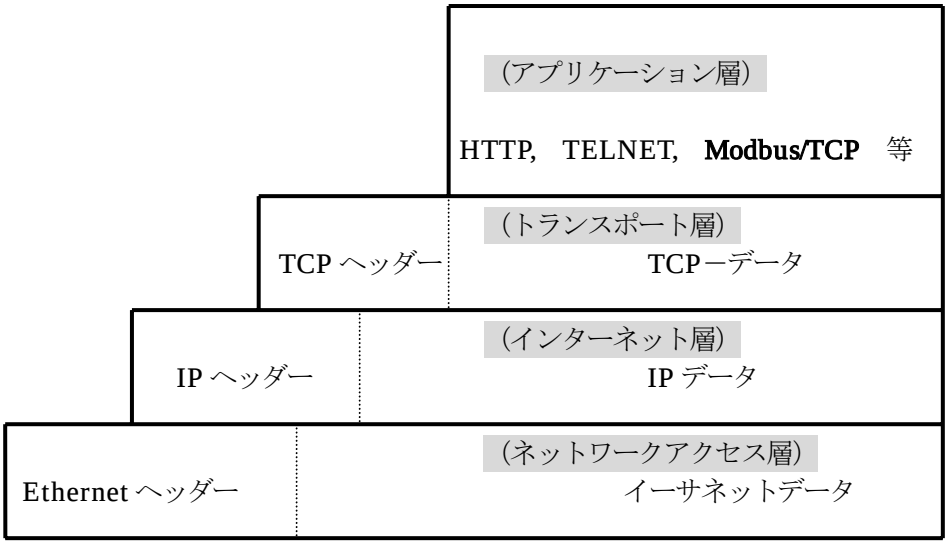
```
static char auchCRCLo[] = {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
```

0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40
};

5.3 Ethernet Modbus／TCP プロトコル

5.3.1 Modbus/TCP プロトコル

イーサネットバスカップラでは TCP/IP のアプリケーション層においてデータの交換行いますので、TCP/IP のトランスポート層以下の機能を考慮する必要はありません。BootP テーブルソフトウェアで書込まれた IP アドレスを指定して通信を確立した後は、HTTP や TELNET 等と同様にアプリケーションレベルでのデータ通信を行います。 通信プロトコルはオープンプロトコルである「**Modbus/TCP**」を使用しており、このアプリケーションプロトコルは**ポート番号５０２**で通信を行います。



5.3.2 Modbus/TCP のデータ構造

アプリケーション層での「Modbus/TCP」のデータは以下のように byte 単位で区切られます。実際の TCP/IP 通信上では各々のバイトデータを１６進数に変換して通信を行います。

Byte No	0	1	2	3	4	5	6	7	8～
データの内容	転送 ID「0」で固定		ﾌﾟﾛﾄｺﾙ ID「0」で固定		Byte6 以降の総ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID 使用せず、「1」で固定	Modbus/TCP の機能コード	機能コードに続く転送データ

＜データ例＞

Byte No	0	1	2	3	4	5	6	7	8～	
Byte データ	0	0	0	0	0	6	1	1	0	8
１６進数(転送ﾃﾞｰﾀ)	00	00	00	00	00	06	01	01	8	

5.3.3 データ読み出しと書き込みの機能分類

スレーブであるワゴ I/O からデータを読出、あるいはデータを書込む動作は、ホスト（パソコン）側から送信される Modbus/TCP 中のコマンド（Byte 7）にある機能コードによって区別します。同一のノード内に入力モジュールと出力モジュールが混在していても、読み出し動作（入力モジュール）と書き込み動作（出力モジュール）はそれぞれ独立した機能として扱います（シリアル Modbus プロトコルと同じ）すなわち**読み出し動作時は入力モジュールのみが対象**となり、**書き込み動作時は出力モジュールのみが対象**となります。尚、750-404 カウンターモジュールについては入力、出力ともに６バイト（４８ビット）のビット幅を占有しますのでご注意ください。

読み出し、書き込み動作は、以下の表のように **8 種類の機能**に分類されます。５－２－４以降に各々の動作の詳細を説明してありますが、表中の**「機能コード」は Modbus/TCP の機能コード（１６進数）に対応**しています。

●ワゴ I/O からデータを読み出す（入力）　　「　」内は１６進数

分類	機　　能	対象モジュール	**機能コード：１６進数** （Byte 7 のデータ）
①	ビット単位で範囲を指定してデータを読みだす。	ディジタル入力モジュール	「０１」または「０２」
②	ワード単位で範囲を指定してデータを読み出す	アナログ入力モジュールおよびディジタル入力モジュール	「０３」または「０４」
③	電源投入以降に実行されたコマンドの回数を読み出す	バスカップラ	「0B」

●ワゴ I/O へデータ書き込む（出力）

分類	機　　能	対象モジュール	機能コード：１６進数
④	単一のビットを指定してデータを書き込む	ディジタル出力モジュール	「０５」
⑤	範囲を指定して複数ビットにデータを書き込む	ディジタル出力モジュール	「０Ｆ」
⑥	単一のワードを指定してデータを書き込む	アナログ出力モジュールおよびディジタル出力モジュール	「０６」
⑦	範囲を指定して複数ワード単位でデータを書き込む	アナログ出力モジュールおよびディジタル出力モジュール	「１０」
⑧	最初の８ビット分の出力データを読む	ディジタル出力モジュール	「０７」

5.3.4 読み出し、書き込みコマンドの送出方法

マスター（ホストコンピューター）とスレーブ（ワゴ I/O）の IP アドレスにて TCP 接続が確立された後は **Port502** 経由で通信を行います。ホストコンピュータからのリクエスト信号を受け取ったワゴ I/O は機能コードで指定された動作を行い、確認のアンサーバックをホストコンピュータへ返信します。

●リクエスト信号（マスター → ワゴ I/O）の例

Byte No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ 1		転送データ 2	
Byte データ	0	0	0	0	0	6	1	1	0	0	0	8
１６進数（転送コード）	00 00		00 00		00 06		01	01	00 00 開始アドレス		00 08 範囲(ビット)	

この部分は 0 で固定します。（未使用）

1 で固定します。（未使用）

機能コードによってこのデータの長さと意味は異なります。この場合は、転送データ 1 が数え始の開始アドレス、データ 2 が開始アドレスからの範囲（ビット数）を示します。

Modbus/TCP の動作機能コードです。例えばこの場合は 1 ですから指定したビットデータの値を読み出すという意味になります。

ユニット ID 以下の転送総バイト数です。

● アンサーバック信号（ ワゴ I/O → マスター）の例

Byte No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ 1	転送データ 2
Byte データ	0	0	0	0	0	4	1	1	1	18
１６進数（転送コード）	00 00		00 00		00 04		01	01	01 データのバイト数	12 データ

アンサーバック信号中のデータもこの部分は未使用で、0 に固定されています。

リクエスト信号で要求された機能コードを確認のために返信します。

転送に含まれるデータのバイト数を示します。転送データ 2 が 1 バイト分であることを意味しています。

ユニット ID 以下の転送バイト数が 4 バイトであることを意味します。

データは 4 ビット分×2 セット（1 バイト）ごとに 2 桁の１６進数を用いて表わします。この場合は上位の 4 ビットが 1 、下位の 4 ビットが 2 ですから、ビット構成に直すと以下のような並びになります。

2 1

○ ● ○ ○ ● ○ ○ ○

(1) (2) (4) (8) (1) (2) (4) (8)

← バスカップラに近い方向

(●：入力 オン)

(○：入力 オフ)

5.3.5　　ワゴ I/O からデータを読み出す

① ビット単位で範囲を指定してデータを読み出す（機能コード　「０１」または「０２」）

このコマンドはビット単位で開始位置を設定、ビット単位で範囲を指定して、その入力値（「１」か「０」また　は「High」か「Low」）を読み出します。リクエスト信号とアンサーバックの信号の例は５－２－４で説明したとおりです。このコマンドは**ディジタル入力モジュールのみが対象**となります。また、ビット単位での最高点数は２５６点（３２バイト）分です。すなわちリクエスト信号の中の開始アドレス（転送データ１）における最高値は2桁１６進数で「FF」です。
アンサーバック信号中のデータ（転送データ２）は８ビット単位（バイト単位）で区切られ、バスカップラに近いディジタル入力モジュールの４ビット分の１６進数値がデータの下位桁によって表され、バスカップラから遠いモジュールの４ビット分の１６進数値が上位桁に表されます。

② 範囲を指定してワード単位でデータを読み出す（機能コード　「０３」または「０４」）

このコマンドは１６ビット単位（ワード単位）で範囲を指定してワゴＩ／Ｏからデータを読み出します。ディジタル入力モジュールが接続されている場合は、アンサーバック信号中の１６進数値の構成は４ビット×２セット（１バイト）を単位として上位と下位２桁ずつの組み合わせで表します。またアナログモジュールが挿入されていた場合はアナログ入力モジュール（１チャンネルあたり１６ビット）が優先されます。すなわちディジタル入力モジュールのデータはアナログデータの後に続くことになります。

図５－２－５

●リクエスト信号（マスター　→　ワゴ I/O）

Byte　No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ 1		転送データ 2	
Byte データ	0	0	0	0	0	6	1	3	0	0	0	2
１６進数（転送コード）	00　00		00　00		00　06		01	03	00　00 **開始アドレス**		00　02 **何ワード分**	

この部分は　０　で固定します。（未使用）

１で固定します。（未使用）

「03」はワード単位で範囲指定してデータを読み出すという機能コードです。

数え始めの開始アドレスから何ワード分読むかを指定します。ここでは 2 ワード分を指定しています。

ユニット ID 以下の転送総バイト数です。

数え始めの開始アドレスをワード単位で指定します。例えば 2 チャンネルのアナログモジュール 1 個が挿入されている場合には、これが優優先されますので 1 ワード目「0000」がアナログ CH1,　2 ワード目が 001 アナログ CH2、その後にディジタル入力ビットが割り当てられます。アナログモジュールのデータを無視して 3 ワード目からディジタル入力データだけを読む場合はここを 02「0002」と設定します。

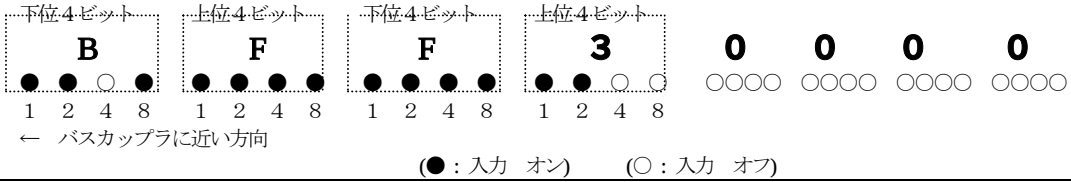
●アンサーバック信号（ワゴ I/O　→　マスター）

Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ1	転送ﾃﾞｰﾀ2	転送ﾃﾞｰﾀ3	転送ﾃﾞｰﾀ4	転送ﾃﾞｰﾀ5
Byte ﾃﾞｰﾀ	0	0	0	0	0	7	1	3	4	63	251	0	0
１６進数（転送コード）	00 00		00 00		00 07		01	03	04 ﾃﾞｰﾀ量	3F FB 読込データ		00 00 読込データ	

ﾕﾆｯﾄID以降のデータ量をByte単位で表します。

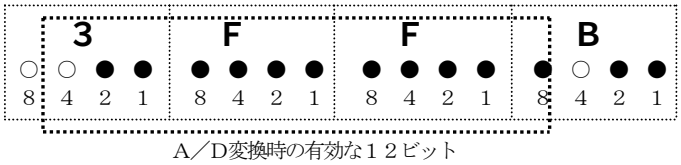
転送データ量（転送ﾃﾞｰﾀ2以降）が４バイト分あることを示しています。

データは４ビット分×２セット（１バイト）ごとに４桁の１６進数（１ワード）を用いて表わします。この場合は下位のワード分が「**３ＦＦＢ**」で、この１６ビット分が下位ワード（レジスター０）になります。これを分解すると以下のようなビット構成になります。仮にすべてがディジタル入力である場合はビットの並びはバスカップラに近い方から以下の並びと同じになります。



(●：入力　オン)　　(○：入力　オフ)

２ＣＨアナログ入力モジュールが挿入されているときは、１チャンネルあたり１６ビットが割り当てられます。下位ワード（レジスタ０）の１２ビット分を使います。「３ＦＦＢ」の場合は、以下のように分解され、１２ビットのうちの１１ビット分がオンの状態を示します。ただし、アナログの場合は下位３ビットは使用せず、通常はビットがＨｉのままになりますので数値は**「３ＦＦＦ」になります。**



A／D変換時の有効な１２ビット

仮に０～１０Ｖのアナログ入力モジュールに適用すると以下のようになります。

$$\frac{2^{11}\times 0+2^{10}\times 1+2^{9}\times 1+2^{8}\times 1+2^{7}\times 1+2^{6}\times 1+2^{5}\times 1+2^{4}\times 1+2^{3}\times 1+2^{2}\times 1+2^{1}\times 1+2^{0}\times 1}{2^{12}}\times 10\text{V}$$

$$= \frac{2047}{4095}\times 10\text{V} = 5\text{V}$$

あるいは、電圧アナログモジュールは　0008（hex）～７FFF（hex）が　０～１０Ｖに相当しますので、３FFB(hex)=16379,　よって

10V÷（7FFF[hex]―0008[hex]）×（３FFB[hex]―00008[hex]）＝

＝１０V÷（32768－８）×(16379―８)＝　５V

尚、各モジュールのビットフォーマットについては各モジュールの説明の項をご覧ください。

③実行されたコマンド回数を読み出す（機能コード「０Ｂ」）

このコマンドはバスカップラ内にあるカウンタのカウント数値を読み出します。バスカップラ内のコマンドカウンタは電源が投入された後、正常に動作が行われたコマンドの回数をカウントしていますので、マスター側から送り出したコマンドの数とこの数をチェックすることによってそのノードとの通信が正常に行われているかどうかを確認することができます。

●リクエスト信号（マスター　→　ワゴ I/O）

Byte　No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード
Byte データ	0	0	0	0	0	2	1	11
１６進数（転送コード）	00 00		00 00		00 02		01	0B

この部分は　０　で固定します。（未使用）

１で固定します。（未使用）

「0B」はバスカップラ内のカウンターの数値を読み出すという機能。コードです。

ユニット ID 以下の転送総バイト数です。

●アンサーバック信号（ワゴ I/O　→　マスター）

Byte　No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ１		転送データ２	
Byte データ	0	0	0	0	0	6	1	11	0	0	2	212
１６進数（転送コード）	0000		0000		0006		01	0B	0000		02　D4 カウンタ数値	

ユニットID以降のデータ量をByte単位で表します。

ステータスをあらわします。常に「0000」です。

電源投入以降のコマンド実行回数が７２４回目であることを示しています。１６ビット、最大６５５３５回までのカウントができます。

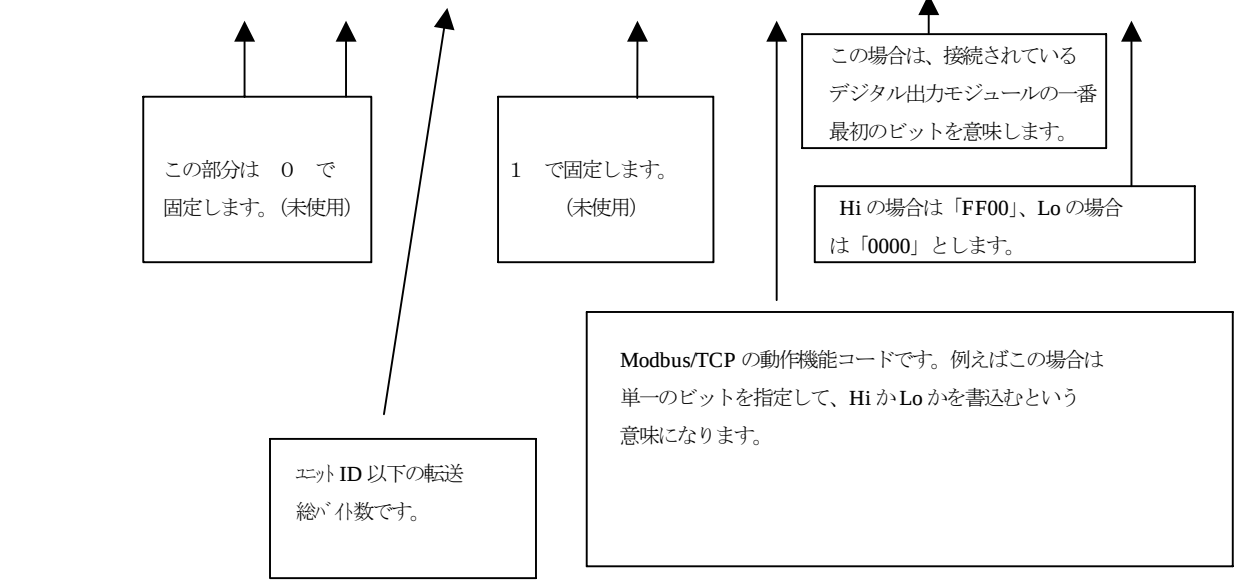
5.3.6　　ワゴ I/O へデータを書き込む

④単一のビットを指定してデータを書き込む（機能コード「０５」）

このコマンドは単一のビットを指定して、ＯＮ／ＯＦＦの出力を実行します。出力の実行は単一のビットですが、設定できる範囲は０００００～００ＦＦまでの２５６ポイントの範囲です。この機能の対象モジュールはディジタル出力モジュールのみです。アナログ出力モジュールは対象外です。

●リクエスト信号（マスター　→　ワゴ I/O）の例

Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ 1		転送ﾃﾞｰﾀ 2	
Byte ﾃﾞｰﾀ	0	0	0	0	0	6	1	5	0	0	255	0
１６進数 （転送コード）	00　00		00　00		00　06		01	05	00　00 開始アドレス		FF　00 Hi/Lo の指示値	



●　アンサーバック信号：（　ワゴ I/O　→　マスター）

Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ 1		転送ﾃﾞｰﾀ 2	
Byte ﾃﾞｰﾀ	0	0	0	0	0	6	1	5	0	0	255	0
１６進数 （転送コード）	00　00		00　00		00　06		01	05	00 00 開始ｱﾄﾞﾚｽ		FF　00 Hi/Lo の指示値	

この機能コードの場合は、動作した内容をそのまま確認のため返信しますので、送信した内容と同じ値がアンサーバック信号として返信されます。

⑤ 範囲を指定して複数ビットにデータを書き込む（機能コード「０F」）

このコマンドは複数のビットを指定して、データを書き込みます。アナログモジュールは対象外です。

●リクエスト信号（マスター　→　ワゴ I/O）の例

Byte　No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ 1		転送データ 2		転送データ 3	転送データ 4	
Byte データ	0	0	0	0	0	9	1	15	0	0	0	16	2	165	240
１６進数（転送コード）	00	00	00	00	00	09	01	0F	00	00	00	10	02	A5	F0
									開始アドレス		範囲		範囲	書込み値	

この部分は　0　で固定します。（未使用）

1　で固定します。（未使用）

書き込み始めのビットのアドレスです。0000(hex)が最初のビットを意味します。

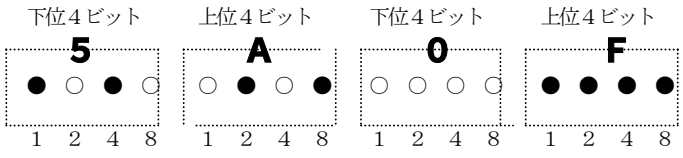
Modbus/TCP の動作機能コードです。この場合は 0F（hex）ですから範囲を指定して複数のビットに書込むという意味になります。

書き込みするビットの範囲をビット数で示します。この場合は 16 ビット分を書込みするという意味です。

ユニット ID 以下の転送総バイト数です。

書き込みする範囲をバイト数で表します。16 ビット分ですから 2 バイトに相当します。

データは４ビット分×２セット（１バイト）ごとに４桁の１６進数（１ワード）を用いて表わします。この場合は指定された２バイト分（１ワード分）が「Ａ５Ｆ０」で、これを分解すると以下のようなビット構成になります。ビットの並びはバスカップラに近い方から以下の並びと同じになります。



←　バスカップラに近い方向

(●：入力　オン)　　(○：入力　オフ)

●アンサーバック信号（ワゴ I/O→マスター）

Byte　No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ 1		転送データ 2		以下省略
Byte データ	0	0	0	0	0	6	1	15	0	0	0	16	
１６進数（転送コード）	00	00	00	00	00	06	01	0F	00	00	00	10	
									開始アドレス		範囲		

アンサーバック信号は上のように転送データ２までの内容を確認のため返信します。

⑥ 単一のワードを指定してデータを書き込む

このコマンドは単一のレジスタ（１６ビット分）に対してデータを書き込むためのコマンドです。例えばアナログ出力モジュールが挿入されている場合にはこれが優先されます。アドレス「０００００」（レジスタ０）がアナログ１CH 目、アドレス「０００１」（レジスタ１）がアナログ２CH 目になります。

●リクエスト信号（マスター　→　ワゴ I/O）

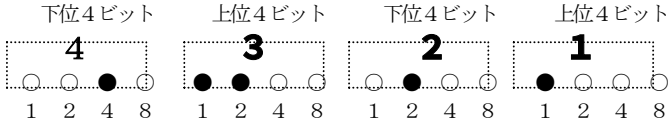
Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ 1		転送ﾃﾞｰﾀ 2	
Byte ﾃﾞｰﾀ	0	0	0	0	0	6	1	6	0	1	18	52
１６進数（転送コード）	00	00	00	00	00	09	01	06	00	01	12	34
									開始ｱﾄﾞﾚｽ		書込み値	

転送 ID、プロトコル ID は「0000」で固定、ユニット ID も１で固定､転送バイトはユニット ID 以下の総バイト数をあらわします。

単一のレジスター（１６ビット分）のみを書込むコマンドは「06」です。

数え始めのスタートアドレスをワード単位で指定します。１６進数で設定します。例えば、２チャンネルのアナログ出力モジュール１個が挿入されているときには、これが優先されますので、この場合は１ワード目「０００００」がアナログ CH1, ２ワード目「０００１」がアナログ CH2, その後にディジタル出力ビットが、バスカップラに近い方から割り当てられます。
　アナログ出力モジュールのデータを無視して３ワード目からディジタル出力モジュールにのみデータを書き込みたい場合はここを「０００２」と設定します。

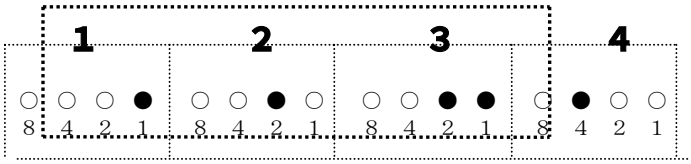
データは４ビット分×２セット（１バイト）ごとに４桁の１６進数（１ワード）を用いて表わしますこのコマンドはレジスタ１個分、すなわち１６ビット分の書き込みしかできませんので、４ビット×２セット×２セット分のデータを以下のように４桁の１６進数で表現します。仮にすべてがディジタル出力モジュールである場合はビットの並びはバスカップラに近い方から以下の並びと同じになります。



← バスカップラに近い方向

(●：入力　オン)　　(○：入力　オフ)

２CHアナログ出力モジュールが挿入されているときは、アドレスを「０００００」に設定してデータを送ると、まずアナログ出力の CH1 にデータが適用されます。アナログ出力１CH あたり１６ビット（１ワード）が割り当てられ、以下のように１２ビット分を使います。例えば１０V の出力モジュールで「１２３４」を出力すると１．４２V になります。



A／D変換時の有効な１２ビット

$$\frac{2^{11}\times 0+2^{10}\times 0+2^{9}\times 1+2^{8}\times 0+2^{7}\times 0+2^{6}\times 1+2^{5}\times 0+2^{4}\times 0+2^{3}\times 0+2^{2}\times 1+2^{1}\times 1+2^{0}\times 0}{2^{12}}\times 10V$$

$$= \frac{582}{4095} \times 10V = 1.42V$$

●アンサーバック信号（ワゴ I/O→マスター）

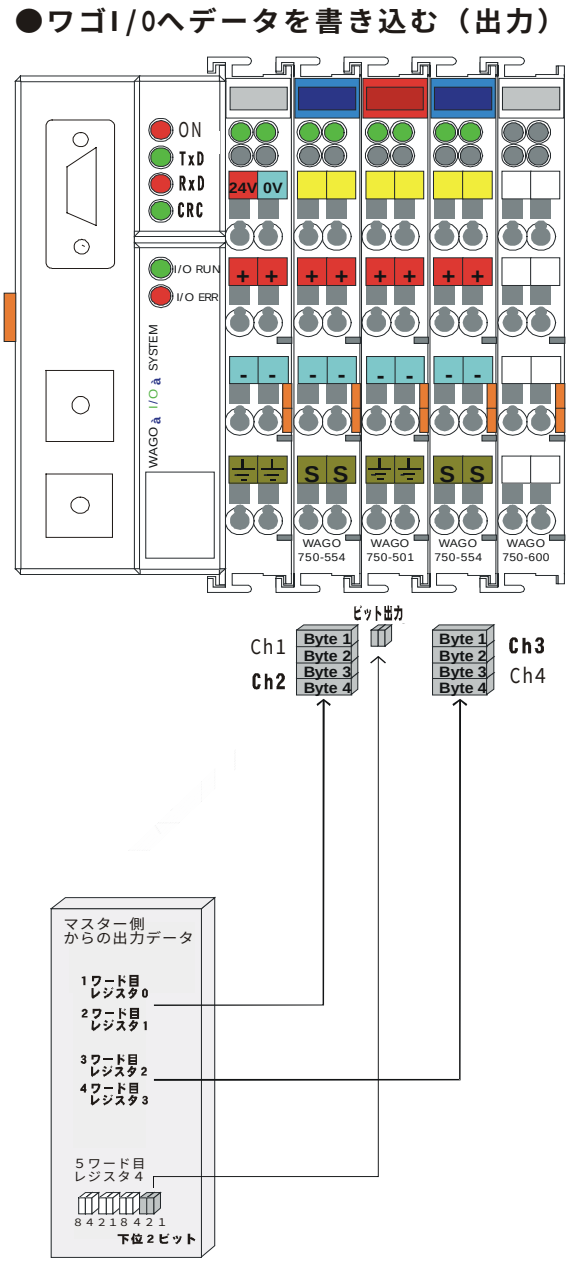
Byte　No	転送 ID		プロトコル ID		転送バイト数		ユニット ID	機能コード	転送データ 1		転送データ 2	
Byte データ	0	0	0	0	0	6	1	6	0	1	18	52
１６進数 （転送コード）	00	00	00	00	00	06	01	06	00	01	12	34

アンサーバック信号は確認の意味でリクエスト信号と同じ内容を返信します。

⑦ 範囲を指定して複数ワード単位でデータを書き込む

このコマンドは複数のレジスタ（１６ビット分×N）に対してデータを書き込むためのコマンドです。アナログ出力モジュールが挿入されている場合にはこれが優先されて割り当てられます。出力データはレジスタの低い方から順番に４桁の１６進数で送出します。

図５－４－１



●リクエスト信号（マスター　→　ワゴI/O）

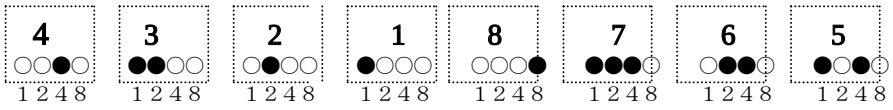
Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ユニット ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ 1		転送ﾃﾞｰﾀ 2		転 送 ﾃﾞｰﾀ 3	転送ﾃﾞｰﾀ 4		転送ﾃﾞｰﾀ 5	
Byte ﾃﾞｰﾀ	0	0	0	0	0	11	1	16	0	0	0	2	4	18	52	86	120
１６進数 (転送コード)	00　00		00　00		00　0B		01	10	00　00 開始 ｱﾄﾞﾚｽ		00　02 範囲 (ﾜｰﾄﾞ)		04 範囲 (ﾊﾞｲﾄ)	12　34 書込み値		56　78 書込み値	

数え始めのスタートアドレスをワード単位で指定します。１６進数で設定します。例えば、２チャンネルのアナログ出力モジュール１個が挿入されているときには、これが優先されますので、この場合は1ワード目「００００」アナログCH1，２ワード目「０００１」がアナログCH2，その後にディジタル出力ビットが、バスカップラに近い方から割り当てられます。
　アナログ出力モジュールのデータを無視して３ワード目からディジタル出力モジュールにのみデータを書き込みたい場合はここを「０００２」と設定します。

出力する範囲をワード単位で設定します。この場合はスタートアドレス「００００」から数えて「０００２」（２ワード分＝３２ビット分）を範囲とするという意味になります。

スタートアドレスから何バイト分を範囲にするかを指定します。この場合は３２ビット分を範囲設定しているので４バイトを設定しています。出力範囲ビットカバーする数値に設定してください。

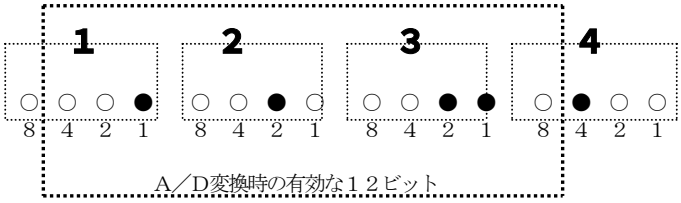
　データは４ビット分×２セット（１バイト）ごとに２桁の１６進数を用いて表わしますこのコマンドはレジスタを複数個分、すなわち１６ビット分×ｎ個の書き込みが可能です。１６ビットごとに４桁の１６進数で置き換えながらデータを作ります。仮にすべてがディジタル出力モジュールである場合はビットの並びはバスカップラに近い方から以下の並びと同じになります。



←　バスカップラに近い方向

(●：入力　オン)　　(○：入力　オフ)

　アナログ出力モジュールが挿入されているときは、アドレスを「００００」に設定してデータを送ると、まずアナログ出力のCH1にデータ
が適用されます。アナログ出力1CHあたり１６ビット（１ワード）が割り当てられ、以下のように１２ビット分を使います。例えば「１２３４」のデータをCH１に送ると、以下のように　Vが出力されます。



$$\frac{2^{11}\times 0+2^{10}\times 0+2^{9}\times 1+2^{8}\times 0+2^{7}\times 0+2^{6}\times 1+2^{5}\times 0+2^{4}\times 0+2^{3}\times 0+2^{2}\times 1+2^{1}\times 1+2^{0}\times 0}{2^{12}}\times 10\mathrm{V}$$

$$= \frac{582}{4095} \times 10\mathrm{V} = 1.42\mathrm{V}$$

●アンサーバック信号（ワゴ I/O→マスター）

Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ 1		転送ﾃﾞｰﾀ 2	
Byte ﾃﾞｰﾀ	0	0	0	0	0	6	1	16	0	0	0	2
１６進数（転送コード）	00	00	00	00	00	06	01	10	00	00	00	02

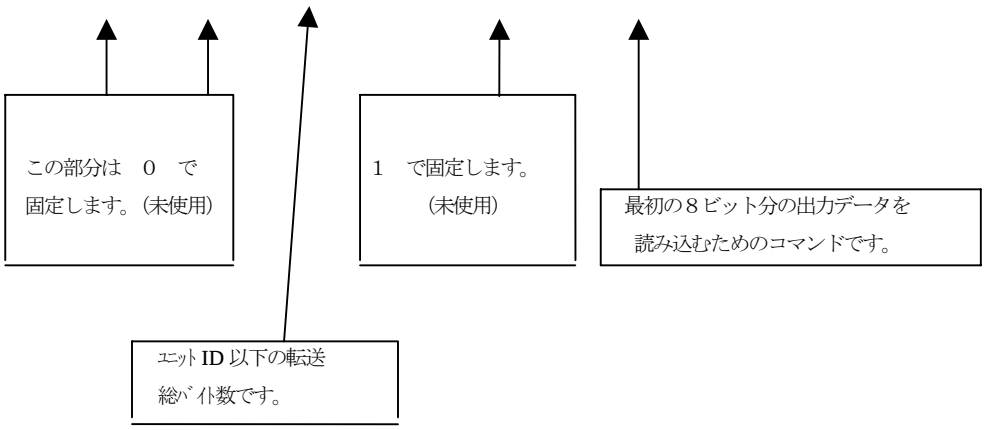
アンサーバック信号は上のように転送データ２までの内容を確認のため返信します。

⑧　最初の８ビット分の出力データを読む

このコマンドは既に出力されている８ビット分の出力データの状態を確認するためのコマンドです。
フィールドバスカップラに一番近い８ビット分のディジタル出力モジュールのみが対象になります。

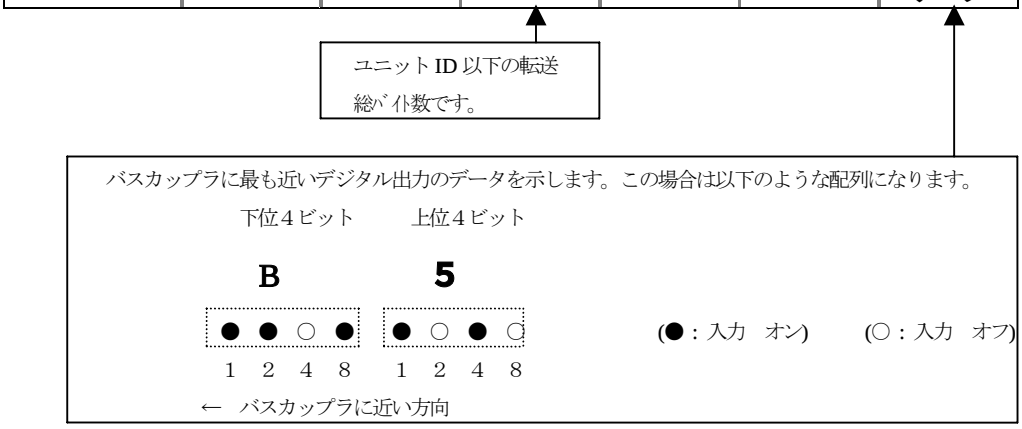
●リクエスト信号（マスター　→　ワゴ I/O）

Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ
Byte ﾃﾞｰﾀ	0	0	0	0	0	2	1	7
１６進数（転送コード）	00	00	00	00	00	02	01	07



●アンサーバック信号：（　ワゴ I/O　→　マスター）

Byte　No	転送 ID		ﾌﾟﾛﾄｺﾙ ID		転送ﾊﾞｲﾄ数		ﾕﾆｯﾄ ID	機能ｺｰﾄﾞ	転送ﾃﾞｰﾀ 1
Byte ﾃﾞｰﾀ	0	0	0	0	0	3	1	7	91
１６進数（転送コード）	00	00	00	00	00	06	01	07	5B データ



6　標準添付　開発支援ソフト（Dynamic　Link　Library）

（Ethernet,　シリアル通信共用）

6.1　システムインストール

添付されている CD―ROM 中の　“DLL”フォルダには自動インストールプログラム（Setup．exe）が含まれていますので以下の手順にしたがってインストールしてください。

① Setup.exe を起動する。

② 「WAGO__IO のセットアップを始めます」→「ＯＫ」をクリックしてください。
次に、「セットアップ開始ボタン」をクリックしてください。
●既に「Wago_io」がインストール済のときはコントロールパネルの「アプリケーションの追加と削除」で古いバージョンのものを削除してください。
●シリアル番号は　**“wago2002”** と入力してください。

③「WAGO_IO のセットアップ完了」→「ＯＫ」をクリックしてインストールが終了します。

④ DLL 本体（WAGO_IO.DLL）は Windows のシステムディレクトリにコピーされます。

Windows　→　System32　→　「Wago_io.dll」

⑤ インストール後のフォルダ構成は以下の通りです。インストールの場所を指定しない場合には Program Files　のフォルダの中に Wago_io　のフォルダが作成されます。

```
Wago_io
Readme.txt
Sample.exe
Config.def
Sample.dep
DeIsL1.isu
DeIsL2.isu
DeIsL3.isu
DeIsL4.isu
    C
        Wago_dll.h      (C 言語用構造体／関数定義ファイル)
        WAGO_IO.lib     (C 言語用ライブラリファイル)
    Vb
        Wago_IO.bas（Visual　Basic 用構造体／関数定義ファイル)
        Sample  (サンプルプログラムのプロジェクト一式)
            Sample.bas
            Sample.frm    :    サンプルプログラムのソースコード
            Sample.frx
            Sample.vbw
            SetPRM.frm
            Config.def
            Sample.vbp
            MSsccprj.scc
        ActiveX
            WagoAxIo.exe
        VbSample
            Form1.frm
            Msccprj.scc
            Project1.vbp
            Project1.vbw
            WagoAxSample.exe
```

6.2　Ｃ言語およびC++での使用について

本ＤＬＬをＣ言語あるいはC++言語から使用する場合は、通常のＷｉｎｄｏｗｓ ＤＬＬと同じに扱えますのでフォルダ [C] 内のヘッダー及びライブラリを使用して下さい。

6.3　ＤＬＬ（Wago_io.dll）の機能

標準添付の開発支援 DLL（Dynamic Link Library）を用いると Microsoft Visual Basic や Visual C にてアプリケーションソフトを開発するときに、モジュールの構成（コンフィグレーション）やビットデータの物理量変換、通信プロトコル等を気にすることなく、アプリケーション開発に専念できます。
アプリケーション起動後は、この Wago_io.dll が実行プログラムと連動して動作しますので、ＶＢＶＣ上では複雑なプログラム記述をする必要がありません。Wago_io.dll が効率良く必要な動作を行います。以下は Wago_io.dll を動作させるための関数一覧です。

	関数名	機能・役割
通信動作	WAGO_OpenCommPort	通信回線をオープンするための関数
	WAGO_CloseCommPort	通信回線をクローズのための関数
ﾓｼﾞｭｰﾙ ｺﾝﾌｨｸﾞﾚｰｼｮﾝ	WAGO_SetConfig	指定ノードのモジュール構成情報を DLL に通知するための関数
データの 入出力	WAGO_GetData	入力モジュールのデータ読み込み処理関数
	WAGO_SetDigitalData	ディジタル出力モジュールへのデータ出力関数
	WAGO_SetAnalogData	アナログ出力モジュールへのデータ出力関数
	WAGO_GetCounter	カウンタモジュールのデータ読み込み関数
	WAGO_SetCounter	カウンタモジュールへのデータ出力関数
	WAGO_GetCommandCoun	コマンド実行カウンタの回数読み取り関数

レジスタを直接読み書きする	WAGO_GetRegData,	レジスタアドレスデータを直接読み出す
	WAGO_SetRegData	レジスタへ直接データを書き込む

＜ハードウェアの設定について＞

6.3.1　「ＲＳ－２３２Ｃ，ＲＳ－４８５の場合」750-312, 750-315, 750-314, 750-316

開発支援 DLL は、RTU（バイナリーモード）で動作するようになっております。バスカップラ内部のビットスイッチ（P4）は ON の状態でお使いください。RTU モードのときは EOF の検出するための時間間隔設定も必要ですが、特殊なプログラムを開発するとき以外は「３フレーム分」に設定してください。
すなわち P1, P2, P3 をすべて「Off」にします。
ビットスイッチの詳細につきましては 「第３章 通信のための設定（シリアル通信）」をご覧ください。

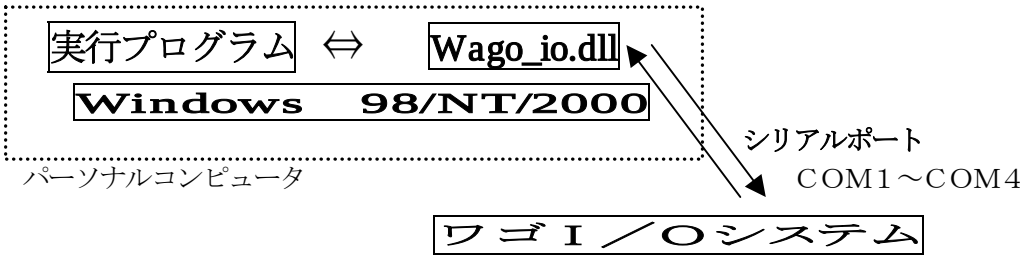
6.3.2　「Ｅｔｈｅｒｎｅｔの場合」750-342, 750-842

開発支援 DLL は、Modbus/TCP で動作するようになっております。バスカップラを利用いただくためには事前に相手となるバスカップラのＩＰアドレスを知る必要があります。バスカップラへのネットワークパラメータの設定（ＩＰアドレス、ネットマスクなど）はＢＯＯＴＰ（ＢＯＯＴＰサーバー）を利用して行います。一度設定したパラメータは保持されますので毎回ＢＯＯＴＰサーバーを用意する必要はありません。

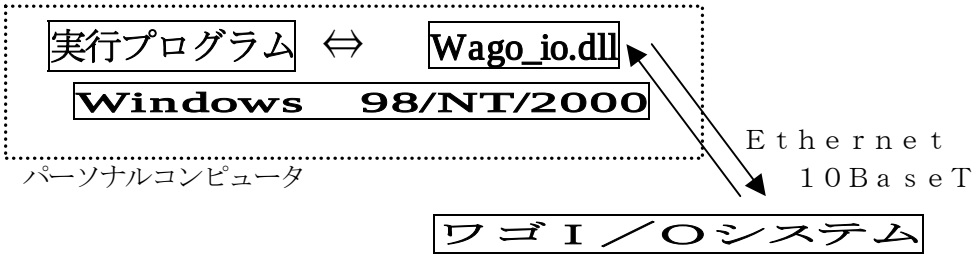
DLL の動作について

Wago_io.dll をパスの設定されているフォルダ（Windows の System フォルダ等)、あるいは実行プログラムと同一のフォルダ内にコピーしてください。プログラム実行時にはコンパイルした実行プログラムと Wago_io.dll が連動してワゴ I/O システム　との通信を実行します。

「ＲＳ－２３２Ｃ，ＲＳ－４８５の場合」



「Ｅｔｈｅｒｎｅｔの場合」



6.4　サンプル　プログラムについて (Sample.exe)

サンプルプログラムは Wago_io フォルダの中にあります。ワゴ I/O をパソコンに接続して簡単な動作テストが実行できます。このソフトは Microsoft Visual　Basic Ver6.0 で作成されており、Sample の中にプロジェクトファイルがすべて入っています。Wago_io.dll の呼出を行っている部分には簡単なコメントが記述されています。Wago_io.dll を使用して開発を行う際の参考にしてください。

① まず、モジュール構成（コンフィグレーション）を決定し、Config.def を開いてコンフィグレーションの内容を記述します。左端は空けずに、バスカップラに近い方から順に、上からモデル名を半角数字で以下のように記載します。Config.def はサンプルプログラム起動時に読み込まれます。すなわちコンフィグレーションの情報をテキストデータで変更できるようにしてあります。

```
750342
750400
750401
750454
750600
  ・
  ・
```

② サンプルプログラムのアイコンが作成されていますのでこれをクリックします。

③ 立ち上がり時に通信設定の画面が出てくるので、パソコンのシリアルポート（COM1～COM4）または Ethernet を選択しシリアルポートならバスカップラで設定されているボーレートを選択

し Ethernet ならＩＰアドレスを入力してください。
750-314/316 の場合フロー制御は「なし」に設定してください。750-312/315 の場合は
使用しているコンバータにもよりますが、「ﾊｰﾄﾞｳｪｱ」に設定してください。

③ プログラム起動後、WAGO I/O ｼｽﾃﾑを接続して簡単な入出力のテストが実行出来ます。

1）データの入力： GetData

・Node ～ 対象とするノードの番号（10 進数）
・Module ～ 指定範囲を決めるときに、その数え始めのモジュール位置を整数で（バスカップラに近い方が１）
＊（注意）指定された場所に入力モジュール以外のモジュール（出力モジュールなど）が設定されているきはデータが正しく得られません。
＊数え始めのモジュールがディジタルモジュールの場合は、ディジタルデータのみのデータ収集になります。
＊アナログとディジタルの両方を対象としたい場合には数え始めのモジュールをアナログ入力モジュールに設定してください。アナログ入力モジュールのデータに引き続いてディジタルモジュールのデータがバスカップラに近い方から取得できます。
・Chl ～ 上で指定されたモジュール内で、何番めのチャンネルから数え始めるか（整数で）。

・Count ～ 上記のチャンネルから同時に入力を行う入力チャンネル数（範囲）を整数で設定。ディジタル入力モジュールから数え始める場合はディジタル入力モジュールだけが対象となり、アナログ入力モジュールから数え始めるとアナログ、ディジタル両方のモジュールが対象となります。（出力モジュールは対象外なのでｶｳﾝﾄしません。）
・Counter ～ 750-404, 750-638 が設定してある場合、このカウンター値を読み出します。

2）データの出力： SetData

・Node ～ 対象とするノードの番号（10 進数）
・Module ～ 指定範囲を決めるときに、その数え始めのモジュール位置を整数で。（バスカップラに近い最初のモジュールが１）
・Chl ～ 上で指定されたモジュール内で、何番めのチャンネルから数え始めるかを整数で。
＊（注意）指定された場所に出力モジュール以外のモジュールが設定されているときは「 Set Error !」のメッセージが出てきます。
・Count ～上記のチャンネルから同時に出力を行う出力チャンネル数（範囲）を整数で設定。アナログ、ディジタルいずれかのモジュールが対象となります。
・Digital/Analog ～ 出力対象をディジタルかアナログかを選択します。
・Counter ～ カウンターの２つのコントロール出力を設定します。

3）実行回数の取得 GetComandCount
電源投入後、正常に実行されたコマンド回数を取得します。

6.5 関数仕様書

6.5.1 モジュール構成（コンフィグレーション情報）通知に関する関数

関数名	**WAGO_SetConfig**
機能	指定ノードのコンフィグレーション情報をＤＬＬに通知する。
引数	**hConn** ～ コネクションハンドル（へのポインタ） **nNodeAdr** ～ ノードアドレスの指定 **pConfig** ～ コンフィグレーション構造体（へのポインタ）
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ Dim hConn As Long 'コネクションハンドル Dim dtCfg As tagConfiguration 'コンフィグレーション構造体の定義 hConn = 0 'ハンドルの初期化 dtCfg.bNotUsedUDP=1 'UDP 通信：0、TCP 通信:1 dtCfg.bErrChk = True 'CRC チェックを有効にする dtCfg.nCnt = 8 '接続されているモジュール数 ' 接続されているモジュールのモデル名称（番号）をバスカップラに ' 近い方から設定（バスカプラも含む） dtCfg.nModelNo(0) = 750342 ' 10Base Ethernet カップラ dtCfg.nModelNo(1) = 750400 ' ２ch ディジタル入力 dtCfg.nModelNo(2) = 750501 ' ２ch ディジタル出力 dtCfg.nModelNo(3) = 750402 ' ４ch ディジタル入力 dtCfg.nModelNo(4) = 750504 ' ４ch ディジタル出力 dtCfg.nModelNo(5) = 750466 ' ２ch アナログ電流入力 dtCfg.nModelNo(6) = 750550 ' ２ch アナログ電圧出力 dtCfg.nModelNo(7) = 750404 ' アップダウンカウンター Call WAGO SetConfig(hConn, 1, dtCfg) ' WAGO__IO.DLL 関数の呼び出し

UDP 通信と TCP 通信　７５０－３４２、７５０－８４２の場合

BNotUsedUDP を設定することによって UDP 通信と TCP 通信を回線毎に設定することができます。これを設定しない場合は自動的に UDP 通信で回線をオープンします。

コネクションハンドル（hConn）変数について

コネクションハンドル（hConn）変数は使用する回線毎に用意する必要があります。複数回線を利用するプログラムの作成ではコネクションハンドル変数を回線分用意する必要があります。
WAGO_GetData,WAGO_SetDigitalData,WAGO_SetAnalogData 関数および WAGO_OpenComPort,WAGO_CloseComPort 関数を実行するときに、どの回線に対して実行するのかを識別する目的でこの変数を使用します。
同時に複数の回線を開く場合には使用する回線の数に相当するコネクションハンドル変数を最初に初期化し、以降は各々の関数実行時に使用します。プログラミング例はこの章の最後に示します。

6.5.2 ポートのオープン・クローズに関する関数

関数名	**WAGO_OpenCommPort**
機能	バスカップラとの通信を行う為にシリアルポートあるいは**Ethernet** のポートをオープンします。
引数	**hConn**～　コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **pProp**～　通信設定構造体（へのポインタ）
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例： ｼﾘｱﾙ通信の場合	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim dtProp As tagCommProp**　　**'** 通信設定構造体を定義します。 **dtProp.nPortNo = 1**　　**'** Ｃｏｍポート番号を指定します。 **dtProp.nParityChk = 0**　　**'** パリティチェックを指定します。 **dtProp.nBaudRate = 19200**　　**'** 通信速度を指定します。 **dtProp.nDataBits = 8**　　**'** データビット長を指定します。 **dtProp.nStopBits = 1**　　**'** ストップビット長を指定します。 **dtProp.nFlowCtrl = 0**　　**'** フローコントロールの指定を行います。 **dtProp.pszHostAddress=vbNull**　‘ **IP**アドレス **Call WAGO_OpenCommPort(hConn, dtProp)** **'** **WAGO_IO.DLL**関数の呼び出しを行います。
使用例： Ethernet の場合	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim dtProp As tagCommProp**　　**'** 通信設定構造体を定義します。 **dtProp.pszHostAddress=“192.192.10.70”‘ IP**アドレス **Call WAGO_OpenCommPort(hConn, dtProp)** **'** **WAGO_IO.DLL**関数の呼び出しを行います。

関数名	**WAGO_CloseCommPort**
機能	シリアルポートあるいは**Ethernet** ポートをクローズします。
引数	**hConn**～　コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定）
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合 ＞ **Call WAGO_CloseCommPort(hConn)**　　**'** **WAGO_IO.DLL** 関数の呼び出しを行います。

6.5.3　　データ入力（取得）に関する関数

関数名	**WAGO_GetData**
機能	入力モジュールのデータ読み込み処理関数。アナログ・ディジタルの両方のモジュールを同時に扱うことができます。
対象 ﾓｼﾞｭｰﾙ	750-400、750-401、750-402、750-403、750-405、750-406、750-408、750-409、750-410 750-412、750-414、750-415、750-452、750-454、750-456、750-461、750-462、 750-465 750-466、750-467、750-468、750-469、および 750-506 の診断データ 750472, 750-474, 750-479, 750-480, 750-476, 750-478
引数	**hConn**　～　コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **nNodeAdr**　～　ノードアドレスの指定 **nMdlNo**　～　指定範囲を決めるときに、その数え始めのモジュール位置を整数で（バスカップラに近い方が１） **nChlNo**　～　上で指定されたモジュール内で、何番めのチャンネルから数え始めるか（整数で）。 **nChlCnt**　～　上記のチャンネルから同時に入力を行う入力チャンネル数（範囲）を整数で設定。アナログ、ディジタル両方のモジュールが対象となります。
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim dtData As tagChlData　　' チャンネルデータ構造体を定義します。** **Dim i　　As Integer　　'** **'** **Call WAGO_GetData(hConn, 1, 1, 1, 6, dtData) ' WAGO_IO.DLL 関数の呼び出し** **' を行います。** **'** **For i = 1 to 6　　'** **　Debug.Print dtData.nData(i)　　' 取得したデータをデバックウィンドウに** **Next i　　' 表示**

1）取得されたデータはディジタル・アナログモジュールに関係無くチャンネル毎に独立した形で得られます。ディジタルモジュールの各チャンネルのデータは ON が「1」、OFF が「0」で表わされます。

2）アナログデータは物理単位そのままの値を表わします。単位については　「V」，「mA」，「℃」がありますが、それぞれのモジュールに対応した物理単位で処理してください。

3）７５０－５０６の診断データについては各チャンネルごとに以下のようなビット構成になっていますが、データは各々のチャンネルごとに０～３の 10 進数で表わすことができます。

	Bit 3	Bit 2	Bit 1	Bit 0
機　　能	Ch2 **出力の診断**		Ch1 **出力の診断**	
正常	0	0	0	0
負荷が接続されていない（または断線）	0	1	0	1
短絡（出力回路のショート）	1	0	1	0
電源電圧が極端に低い	1	1	1	1

4）それぞれのアナログ入力モジュールの読み取り分解能は以下のとおりです。

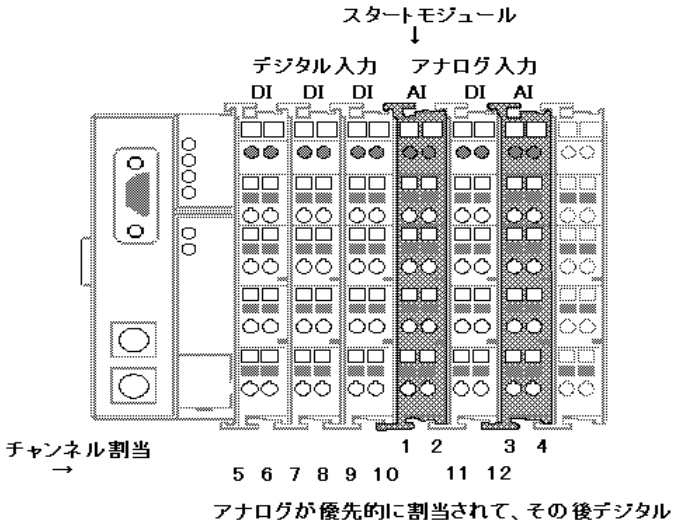
モデル名	出力範囲	有効な最小単位
７５０－４６７・４６８	0～10　V	10／4096＝　0.0024　V
７５０－４５２・４６５	0～20　mA	20／4096＝　0.0048　mA
７５０－４５４・４６６	4～20　mA	16／4096＝　0.0039　mA
７５０－４５６	－10～　+10　V	10／4096=　0.0024　V
７５０－４７２	0～20　mA	20／32768＝0.0006　mA
７５０－４７４	4～20　mA	16／32768＝0.0005　mA
７５０－４７６	0～10　V	10／32768＝ 0.0003　V
７５０－４７８	－10～　+10　V	10／32768＝ 0.0003　V

５） ｎMdlNo＜モジュール番号＞設定について

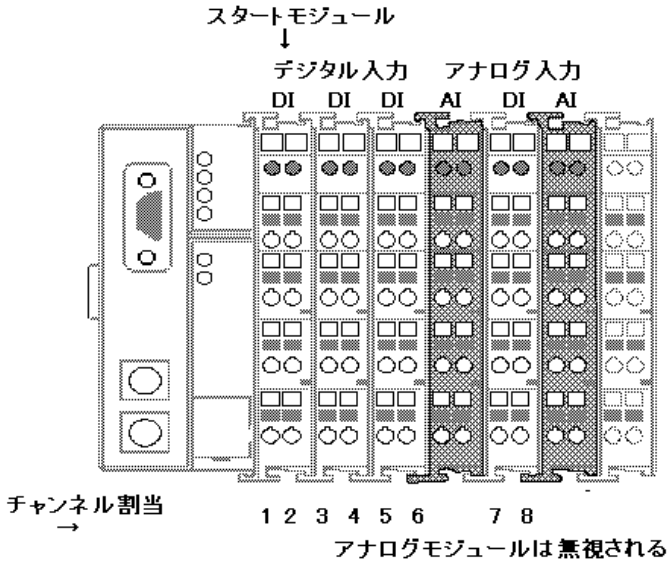
開始するモジュール番号は入力モジュールでなければなりません。入力モジュールとしてはデジタル、アナログモジュール混在していてもかまいませんが、スタートチャンネル（ｎMdlNo）はアナログモジュールにしてください。
デジタルモジュールがアナログモジュールの前（バスカップラに近いほう）に設定されていても、すべてのアナログ入力モジュールの後にデジタルチャンネルが割り当てられます。

●（下図は nMdlNo=４、nChlNo=1、nChlCnt=12　場合のチャンネル割り当て例）

＊　スタートモジュール（ｎMdlNo）をデジタルモジュールに設定すると、デジタル入力モジュールだけが入力の対象となります。



●（下図は nMdlNo=１、nChlNo=1、nChlCnt=8　場合のチャンネル割り当て例）



6.5.4 ディジタル出力に関する関数

関数名	**WAGO_SetDigitalData**
機能	ディジタル出力モジュールへのデータ出力処理関数です。複数のディジタル出力モジュールを同時に扱うことが可能です。
対象 ﾓｼﾞｭｰﾙ	**750-501、750-502、750-504、750-506、750-509、750-516、750-512、750-513、750-514**
引数	**hConn** ～ コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **nNodeAdr** ～ ノードアドレスの指定 **nMdlNo** ～ 指定範囲を決めるときに、その数え始めのモジュール位置を整数で（バスカップラに近い方が１） **nChlNo** ～ 上で指定されたモジュール内で、何番めのチャンネルから数え始めるか（整数で）。 **nChlCnt** ～ 上記のチャンネルから同時に出力を行う出力チャンネル数（範囲）を整数で設定。但しディジタルモジュールだけが対象となります。
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim dtData As tagChlData** ' チャンネルデータ構造体を定義します。 **Dim i As Integer** ' ' **For i = 1 to 6** ' **dtData.nData(i)= i Mod 2** ' 出力データ設定 **Next i** ' **Call WAGO_SetDigitalData(hConn, 1, 2, 1, 6, dtData)** ' **WAGO_IO.DLL** 関数の ' 呼び出しを行います。

6.5.5　　アナログ出力に関する関数

関数名	**WAGO_SetAnalogData**
機能	アナログ出力モジュールへのデータ出力処理を行います。複数のアナログ出力モジュ ールを同時に処理することができます。
対象 ﾓｼﾞｭｰﾙ	**750-550、750-552、750-554、750-556**
引数	**hConn**　～　コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） nNodeAdr　～　ノードアドレスの指定 nMdlNo　～　指定範囲を決めるときに、その数え始めのモジュール位置 を整数で（バスカップラに近い方が１） nChlNo　～　上で指定されたモジュール内で、何番めのチャンネルから 数え始めるか（整数で）。 nChlCnt　～　上記のチャンネルから同時に出力を行う出力チャンネル数（範囲） を整数で設定。但しアナログモジュールだけが対象となります。
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim dtData As tagChlData　' チャンネルデータ構造体を定義します。** **'** **dtData.nData(1) = 3.1　　' 出力データ設定　＊** **dtData.nData(2) = 4.05　　' 出力データ設定　＊** **Call WAGO_SetAnalogData(hConn, 1, 2, 1, 2, dtData)　' WAGO_IO.DLL 関数の** **' 呼び出しを行います。**

１）それぞれのモジュールの設定範囲は以下のとおりです。プログラム記述上では数値分解能についての制約はありませんが、アナログモジュールの分解能は１２ビットですので実際に有効な最小単位は以下のとおりです。

モデル名	出力範囲	有効な最小単位
７５０－５５０	０～10　V	10／4096＝　0.0024　V
７５０－５５２	０～20　mA	20／4096＝　0.0048　mA
７５０－５５４	４～20　mA	16／4096＝　0.0039　mA
７５０－５５６	－10～　+10　V	10／4096=　0.0024　V

6.5.6　　カウンターモジュール取扱いの関数

関数名	**WAGO_GetCounter**
機能	カウンターモジュールのデータ（カウンタ値）読み込み処理
対象 ﾓｼﾞｭｰﾙ	750-404、750-638
引数	**hConn**　～　コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **nNodeAdr**　～　ノードアドレスの指定 **nMdlNo**　～　カウンターモジュールの位置（バスカップラに近い方が１、そこから数えはじめて整数で） **nChlNo**　～　データ取得開始チャンネル番号 **nChlCnt**　～　データ取得チャンネル数（カウンターモジュールが複数個設定されている場合）
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim dtData As tagCntrData　　' カウンターデータ構造体を定義します。** **'** **Call WAGO_GetCounter(hConn, 1, 7, 1, 1, dtData)　' WAGO_IO.DLL 関数の** **' 呼び出しを行います。** **Debug.Print dtData.nCtrl(1); ":"; dtData.nData(1) ' 取得したデータを** **' デバックウィンドウに** **' 表示**

＊カウンターはアップ／ダウンカウントが行えます。変数の型は長整数型を使用しています。

７５０－４０４の場合

読み取り値は　－2,147,483,648　～　＋2,147,483,647 の整数値になります。

７５０－６３８の場合

読み取り値は　０　～　＋６５５３６の整数値になります。

＊　***750-312, 750-314, 750-315, 750-316 については、カウンターモジュールは対応しておりません。***

関数名	**WAGO_SetCounter**
機能	カウンターモジュールへのカウンタリセット、カウンタ値設定
対象 ﾓｼﾞｭｰﾙ	750-404, 750-638
引数	**hConn** ～ コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **nNodeAdr** ～ ノードアドレスの指定 **nMdlNo** ～ カウンターモジュールの位置（バスカップラに近い方が１、そこから数えはじめて整数で） **nChlNo** ～ データ取得開始チャンネル番号（１を設定してください。） **nChlCnt** ～ データ取得チャンネル数（カウンターモジュールが複数個設定されている場合）
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの例＞ **750-404** の場合 **Dim dtData As tagCntrData** ' カウンターデータ構造体を定義します。 ' **dtData.nCtrl(1) = 32** '制御バイト値の設定 **dtData.nData(1) = 100** 'カウンタ値の設定 **Call WAGO_SetCounter(hConn, 1, 7, 1, 1, dtData)** ' **WAGO_IO.DLL** 関数の ' 呼び出しを行います。 **750-638** の場合 **Dim dtData As tagCntrData** ' カウンターデータ構造体を定義します。 ' **dtData.nCtrl(1) = 32** '制御バイト値の設定（**Ch1**） **dtData.nData(1) = 100** 'カウンタ値の設定　（**Ch1**） **dtData.nCtrl(2) = 32** '制御バイト値の設定（**Ch2**） **dtData.nData(2) = 200** 'カウンタ値の設定　（**Ch2**） **Call WAGO_SetCounter(hConn, 1, 7, 1, 1, dtData)** ' **WAGO_IO.DLL** 関数の ' 呼び出しを行います。

750-404 の場合

1）750-404 の制御バイトの内容については以下のとおりです。例えばカウンタ値の設定、リセットするときにはビット５を「1」にしますので 10 進数で３２を設定します。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 **不使用**	0 **不使用**	**カウンター値設定、リセット**	**カウンター動作停止**	**出力 O2 をアクティブ（Hi）にする**	**出力 O1 をアクティブ(Hi)にする**	**0** **不使用**	**0** **不使用**

2）カウンターはアップ／ダウンカウントが行えます。マイナス値も考慮して変数の型は長整数型を使用しています。設定できる数値の範囲は　-2,147,483,648　～　2,147,483,647 の整数値になります。

750-638 の場合

1）出力データ中の制御バイトは下記の各ビットで構成されます。（Ch1, Ch2 共通）
例えばカウンタリセットの場合は Bit 5 を「1」にしますので 10 進数で３２を設定します。

2）アップダウンの変更は Bit 1 を変更することによって動作変更します。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	設定、リセット(1) 動作開始(0)	ｶｳﾝﾀｰ停止(1) 動作開始(0)	0 不使用	0 不使用	ﾀﾞｳﾝｶｳﾝﾄ(1) ｱｯﾌﾟ ｶｳﾝﾄ(0)	ｵｰﾊﾞｰﾌﾛｰでｽﾄｯﾌﾟ (1) ｵｰﾊﾞｰﾌﾛｰでも動作(0)

3）カウンターはアップ／ダウンカウントが行えます。マイナス値も考慮して変数の型は
長整数型を使用ています。設定できる数値の範囲は　０　～　+６５５３５
の整数値になります。

4）制御バイトでカウンターの動作を変更、設定する場合には必ず 2ch を同時に設定してください。片一方のチャンネルだけを設定、リセットすることはできません。

6.5.7　コマンド回数確認のための関数

関数名	WAGO_CommandCount
機能	指定されたノードのバスカップラから実行されたコマンドの回数を読み出す （正常に行われた回数だけがバスカップラのメモリカウンタにカウントされます。 バスカップラの電源をオフにするとこのメモリカウンタはリセットされます。）
引数	**hConn**　～　コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **nNodeAdr**　～　ノードアドレス
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	＜ＶｉｓｕａｌＢａｓｉｃの場合＞ **Dim cnt As Long**　　' ' **cnt = WAGO_GetCommandCount(hConn, 1)** ' WAGO_IO.DLL 関数の呼び出しを行います。 **Debug.Print cnt**　　' 取得したデータをデバッグウインドウに表 ' 示。

6.5.8 レジスタの読み書き関数

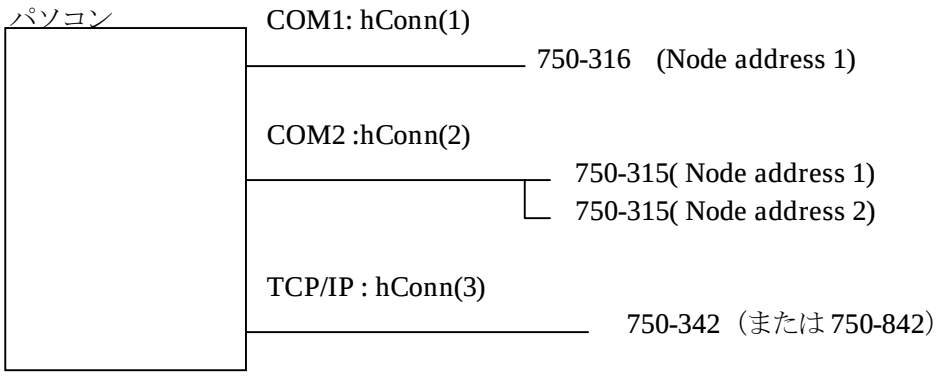
関数名	WAGO_GetRegData, WAWAGO_SetRegData
機能	指定されたレジスタのデータを読み出し、書込み
引数	**hConn** ～ コネクションハンドル（**WAGO_SetConfig** 関数で取得したものを指定） **nNodeAdr** ～ ノードアドレス **nOffset** ～ **16 進数で読み書きする開始アドレスを指定** **nChlCnt** ～ **何ワード分データを読み書きするか（整数）**
戻り値	処理結果 **0:** 正常 **-1:** 異常
使用例	**例) 0x2030 から 64 ワード読むのであれば** **confData.nCnt = 1** **confData.nModelNo(0) = 750842** **nNodeAdr = 0** **WAGO_SetConfig(hConn, nNodeAdr, confData)** **commData.pszHostAddress = "10.10.10.10"** **WAGO_OpenPort(hConn, commData)** **Dim nOffset As Integer** **Dim nChlCnt As Integer** **Dim nData(64) As Integer** **nOffset = &H2030** **nChlCnt = 64** **WAGO_GetRegData(hConn, nNodeAdr, nOffset, nChlCnt, nData(0))** **これでnData( )にデータが返ります。**

●レジスタのアドレス割り当てはモジュールを接続したときにその順番で決定されます。
詳細については４章をごらんください。

6.6 VB プログラムサンプル

コネクションハンドル（hConn）変数は使用する回線毎に用意する必要があります。複数回線を利用するプログラムの作成ではコネクションハンドル変数を回線分用意してそれぞれWAGO_SetConfig関数および WAGO_OpenComPort,WAGO_CloseComPort 関数を呼び出す必要があります。以下に VB でのプログラミング例を示します。

以下の例は２つの COM ポートと TCP/IP のポートを同時に開いて関数を実行する VB のプログラミング例です。




```
Dim  hConn(3)  As  Long

Private Sub Form_Load()
    Call  MyWagoSetConfig
    Call  MyWagoOpen
    Call  MyWagoGetData
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Call  MyWagoClose
End Sub

Private Sub MyWagoConfig()
    Dim  dtCfg      As  tagConfiguration

    hConn(1) = 0                    ‘コネクションハンドルの初期化
    hConn(2) = 0                    ‘コネクションハンドルの初期化
    hConn(3) = 0                    ‘コネクションハンドルの初期化

    ‘ COM ポートに一つの 750-316 を接続する
    dtCfg.bErrChk=True              ‘  エラーチェックを True にする。
    dtCfg.nCnt = 2
    dtCfg.ModelNo(0) = 750316       '   RS-232C バスカップラ
    dtCfg.ModelNo(1) = 750400       '   2ch  ディジタル入力
    Call  WAGO_SetConfig(hConn(1), 1, dtCfg)
```

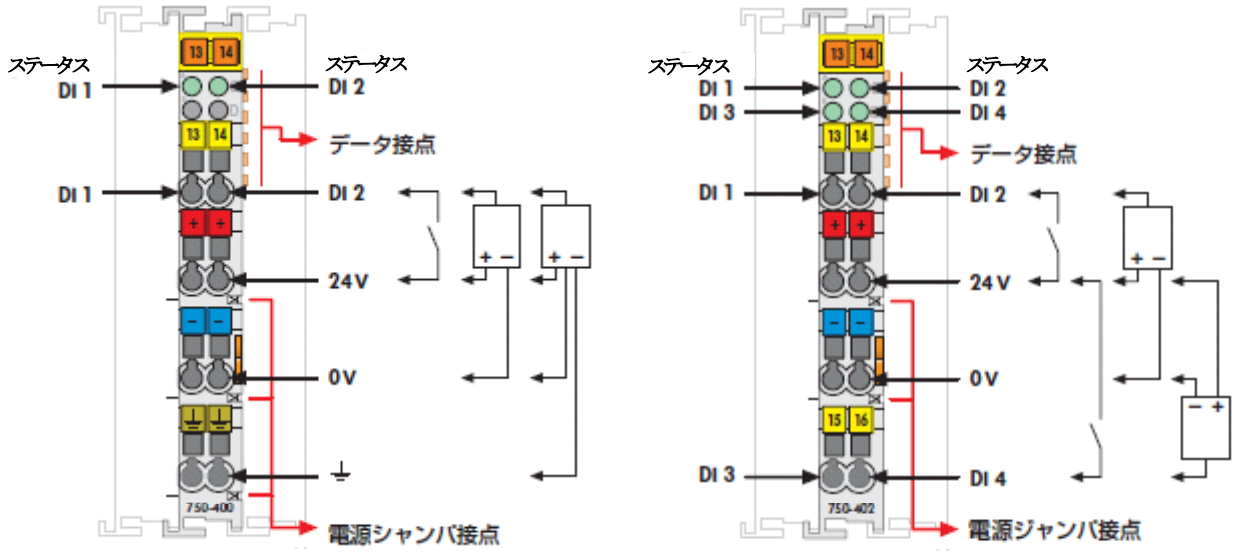
```
    ‘COM ポートに 2 つの 750-315 を接続して２局マルチドロップにする。２台のノード識別はノード
    ‘アドレスで行っている。
    dtCfg.bErrChk=True              ‘ エラーチェックを True にする。
    dtCfg.nCnt = 2
    dtCfg.ModelNo(0) = 750315       '  RS-485 バスカップラ
    dtCfg.ModelNo(1) = 750400       '  2ch  ディジタル入力
    Call  WAGO_SetConfig(hConn(2), 1, dtCfg)

    dtCfg.bErrChk=True              ‘ エラーチェックを True にする。
    dtCfg.nCnt = 2
    dtCfg.ModelNo(0) = 750315       '  RS-485 バスカップラ
    dtCfg.ModelNo(1) = 750400       '  2ch  ディジタル入力
    Call  WAGO_SetConfig(hConn(2), 2, dtCfg)

    ‘ Ｅｔｈｅｒｎｅｔバスカプラ
    dtCfg.bNotUsedUDP=1                ‘ UDP 通信の場合：０、  TCP 通信の場合：１
    dtCfg.nCnt = 3
    dtCfg.ModelNo(0) = 750342       '  Ethernet バスカップラ
    dtCfg.ModelNo(1) = 750400       '  2ch  ディジタル入力
    dtCfg.ModelNo(1) = 750501       '  2ch  ディジタル出力
    Call  WAGO_SetConfig(hConn(3), 0, dtCfg)
End Sub

Private Sub MyWagoOpen()
    Dim dtProp As tagCommProp       ' 通信設定構造体を定義します。

    ‘ ＲＳ－２３２C接続
    dtProp.nPortNo = 1              ' Ｃｏｍポート番号を指定します。
    dtProp.nParityChk = 0           ' パリティチェックを指定します。
    dtProp.nBaudRate = 19200        ' 通信速度を指定します。
    dtProp.nDataBits = 8            ' データビット長を指定します。
    dtProp.nStopBits = 1            ' ストップビット長を指定します。
    dtProp.nFlowCtrl = 0            ' フローコントロールの指定を行います。
    dtProp.pszHostAddress = vbNull  ' Ethernet Host or IP Address
    Call WAGO_OpenCommPort(hConn(1), dtProp)  ' WAGO_IO.DLL関数の呼び出しを行います。

    ‘ ＲＳ－４８５で２局マルチドロップ接続
    dtProp.nPortNo = 4              ' Ｃｏｍポート番号を指定します。
    dtProp.nParityChk = 0           ' パリティチェックを指定します。
    dtProp.nBaudRate = 19200        ' 通信速度を指定します。
    dtProp.nDataBits = 8            ' データビット長を指定します。
    dtProp.nStopBits = 1            ' ストップビット長を指定します。
    dtProp.nFlowCtrl = 0            ' フローコントロールの指定を行います。
    dtProp.pszHostAddress = vbNull  ' Ethernet Host or IP Address
    Call WAGO_OpenCommPort(hConn(2), dtProp)  ' WAGO_IO.DLL関数の呼び出しを行います。
```

```
    ‘ Ｅｔｈｅｒｎｅｔ接続
    dtProp.pszHostAddress = “10.2.3.4”        ' Ethernet Host or IP Address
    Call WAGO_OpenCommPort(hConn(3), dtProp)  ' WAGO_IO.DLL関数の呼び出しを行います。
End Sub

Private Sub MyWagoGetData()
    Dim dtData As tagChlData              ' チャンネルデータ構造体を定義します。
    Dim i      As Integer                 '
    Dim j      As Integer                 '
                                          '
    For I = 1 to 3
        Call WAGO_GetData(hConn(i), 1, 1, 1, 2, dtData) ' WAGO_IO.DLL 関数の呼び出しを行います。
        For j = 1 to 2                    '
            Debug.Print dtData.nData(j)   ' 取得したデータをデバックウィンドウに
        Next j                            ' 表示
    Next i
End Sub

Private Sub MyWagoClose()
    Call WAGO_CloseCommPort(hConn(1))     ' WAGO_IO.DLL関数の呼び出しを行います。
    Call WAGO_CloseCommPort(hConn(2))     ' WAGO_IO.DLL関数の呼び出しを行います。
    Call WAGO_CloseCommPort(hConn(3))     ' WAGO_IO.DLL関数の呼び出しを行います。
End Sub
```

以上

7　デジタル入出力モジュール

7.1　デジタル入力（24V DC; 120V AC, 230V AC, 48V DC）
型番 750-400～415



技術説明

各 I/O モジュールにはそれぞれの動作電圧用として、電源の供給は連続に接続された端子部を通して行われます。I/O モジュールを DIN レール上に嵌め込むと、各モジュール間で電源接続が自動的に行われます。

注意：
一部のモジュール（例：4 チャンネルのデジタル入力モジュール）には一番下の部分の電源ジャンパー接点が用意されていません。したがって、全部で 3 個の電源ジャンパー接点が必要なモジュール（例：2 チャンネルのデジタル入力モジュール）は、3 個の電源ジャンパー接点が用意されていないモジュール（例：4 チャンネルのデジタル入力モジュール）の右側に接続することはできません。　2 チャンネルのデジタル入力はすべて 4 線式デバイス対応になっており、4 線のセンサを V+、0V、グラウンドおよび信号の各端子に直接結線することが可能です。

モジュール 750-408 および 750-409 はネガティブ・スイッチング（負方向接続）用のモジュールです。

750-410 と 750-411 の各モジュールには 2 線式の近接スイッチなどを結線することができます。
ノイズの除去とスイッチのチャッタリング対策を目的として、5V、24V および 48V の各バージョンのデジタル入力には抵抗とコンデンサで構成した RC フィルター回路が直列に結線されています。この RC フィルターの時定数として 3.0ms と 0.2ms のものがあります。

バス動作におけるビット数値割当ては、バスカップラに近い方から始まって左から右へ順番に指定されます。ノード／ステーション内部に取付ける種々の異なる I/O モジュールの位置は順番を変えることによってビット割り当てが決まります。ただしアナログ入力モジュールや特殊モジュールが存在する場合には、これらのアナログ・特殊モジュールが優先的にビット割り当てが行われます。（詳しくは第 4 章をご覧ください。）

技術データ

型番　750-	400	401	402	403
入力数	2		4	
入力フィルターの時定数	3ms	0.2ms	3ms	0.2ms
定格電圧	24V DC（-15%/+20%）			
信号電圧（0）	-3V～+5V DC（std. EN 61131 Type 1）			
信号電圧（1）	15V～30V DC（std. EN 61131 Type 1）			
消費電流（内部）	2.5mA（最大）		5mA（最大）	
入力電流（フィールド側からの）	5mA（平均）			
絶縁耐圧	500V（システム－電源　間）			
内部ビット幅	2		4	
動作温度範囲	0°C～+55°C			
電線接続	ケージクランプ；0.08～2.5mm^2			
電線むき長さ	8-9mm			
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）			

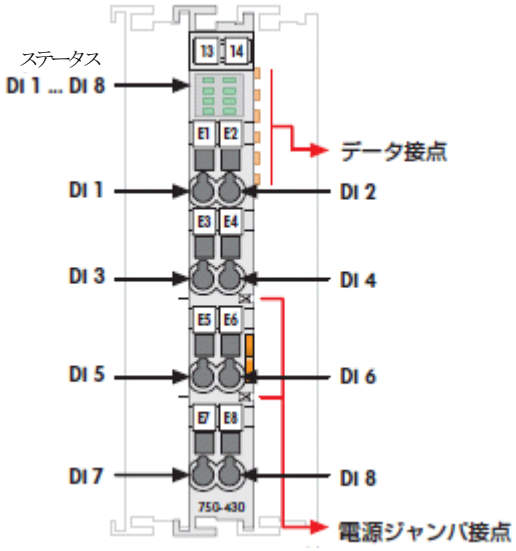
型番　750-	405	406	410*	411*
入力数	2			
入力フィルターの時定数	10ms		3ms	0.2ms
定格電圧	230V AC (-15%/+10%)	120V AC (-15%/+10%)	24V DC (-15%/+20%)	
信号電圧（0）	0V～40V AC	0V～20V AC	-3 V～+5V DC	
信号電圧（1）	79V～253V AC	79V～132V AC	11V～30V DC	
消費電流（内部）	2mA		2.5mA（最大）	
入力電流（フィールド側からの）	6.5mA（平均）	4.5mA（平均）	8mA（平均）	
絶縁耐圧	4kV（システム－電源　間）		500V（システム－電源　間）	
内部ビット幅	2			
動作温度範囲	0°C～+55°C			
電線接続	ケージクランプ；0.08～2.5mm^2			
電線むき長さ	8-9mm			
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）			

*）2 線の近接スイッチ、最大値 2mA の負荷なしの電流

型番　750-	408	409	412	413
入力数	4		2	
入力フィルターの時定数	3ms	0.2ms	3ms	0.2ms
定格電圧	24V DC（-15%/+20%）		48V DC（-15%/+20%）	
信号電圧（0）	15V～30V DC		-6V～+10V DC	
信号電圧（1）	-3V～+5V DC		34V～60V DC	
消費電流（内部）	10mA（最大値）		5mA（最大値）	
入力電流（フィールド側からの）	5mA（平均）		3.5mA（平均）	
絶縁耐圧	500V（システム－電源）			
内部ビット幅	4		2	
動作温度範囲	0°C～+55°C			
電線接続	ケージクランプ；0.08～2.5mm^2			
電線むき長さ	8-9mm			
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）			

型番　750-	414	415
入力数	4	
入力フィルターの時定数	0.2ms	20ms
定格電圧	5V DC	24V AC/DC (-15%/+20%)
信号電圧（0）	0～0.8V DC	-3V～+5V DC 0V～+5V AC
信号電圧（1）	2.4V～5V DC	11V～30V DC 10V～27V AC
消費電流（内部）	5mA	10mA
入力電流（フィールド側からの）	50µA（平均）	7.5mA DC 7.6mA～9.5mA AC
絶縁耐圧	500V（システム－電源　間）	500V（システム－電源　間） 50V（ﾁｬﾝﾈﾙｰﾁｬﾝﾈﾙ　間）
内部ビット幅	4	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm^2	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	

7.2　8 チャンネル デジタル入力（DC 24V/3.0ms/0.2ms、正方向スイッチ）
型番 750-430, 431



技術説明

各 I/O モジュールにはそれぞれの動作電圧用として、電源の供給は連続に接続された端子部を通して行われます。I/O モジュールを DIN レール上に嵌め込むと、各モジュール間で電源接続が自動的に行われます。

注意：
モジュール 750-430 および 750-431 はポジティブ・スイッチング（正方向接続）用のモジュールです。
各々の端子はコモン電源の＋側（＋24V 側）と導通させることによって信号が入力されます。

配線時には、I/O システムに供給する DC24V のコモン電源＋側の端子を外部で準備してください。

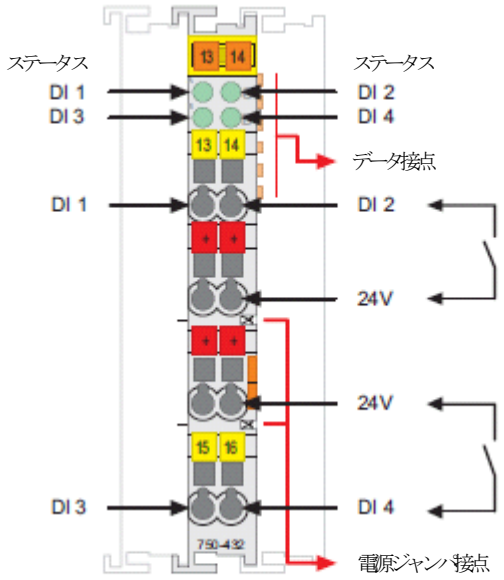
ノイズの除去とスイッチのチャッタリング対策を目的として、入力には抵抗とコンデンサで構成した RC フィルター回路が直列に結線されています。この RC フィルターの時定数として 3.0ms と 0.2ms のものがあります。

バス動作におけるビット数値割当ては、バスカップラに近い方から始まって左から右へ順番に指定されます。ノード／ステーション内部に取付ける種々の異なる I/O モジュールの位置は順番を変えることによってビット割り当てが決まります。ただしアナログ入力モジュールや特殊モジュールが存在する場合には、これらのアナログ・特殊モジュールが優先的にビット割り当てが行われます。（詳しくは第 3 章をご覧ください。）

技術データ

型番　750-	430	431
入力数	8	
入力フィルターの時定数	3ms	0.2ms
定格電圧	24V DC（-15%/+20%）	
信号電圧（0）	-3V～+5V DC	
信号電圧（1）	15V～30V DC	
消費電流（内部）	17mA（最大）	
入力電流（フィールド側からの）	2.8mA（平均）	
絶縁耐圧	500V（システム／電源間）	
内部ビット幅	8	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	

7.3　4 チャンネル デジタル入力（DC 24V／3.0ms／0.2ms、2 線式）
型番 750-432, 433



技術説明

750-432 および 750-433 デジタル入力モジュールは、センサなどのフィールド機器からの制御用信号の入力に用いられます。このモジュールには 4 点の入力チャンネルがあり、2 線式のセンサを 4 個まで直接モジュールにつなぐことができます。
2 線式センサの接続は 24V 端子と DI1、DI 2、DI 3、DI 4 の各端子間で行います。
3 線式センサをこのモジュールにつなぐ場合は、0V 電源の供給用に電源端子拡張モジュール（750-604 または 750-614）を用意する必要があります。

個々の入力チャンネルは時定数が 3.0ms（750-432）または 0.2ms（750-433）の RC ノイズ除去フィルタが付いています。
入力チャンネルの状態はステータス LED で表示されます。
内部回路にはオプトカプラがあり、内部バスとフィールド側との間を電気的に絶縁するのに用いられています。

フィールドバス内ノードを設計するとき、入力モジュールはどんな位置でもコンフィグレーションできます。モジュールのタイプ毎にグループ化する必要はありません。

入力モジュール用の 24V フィールド電源は、隣のモジュールまたは電源入力モジュールから取り出します。フィールド機器用電源は、各 I/O モジュールの電源ジャンパ接点を介して自動的に供給されます。

注意
内部電源ジャンパ接点の最大電流容量は 10A です。システムを構築するときは、各モジュールの消費電流の合計が最大電流容量を超えないようにすることが重要です。もしこの許容量を超えるケースがあった場合は電源入力モジュールを追加しなければなりません。

このモジュールは WAGO-I/O-SYSTEM の全てのバスカプラ／コントローラと共に使用することができます。

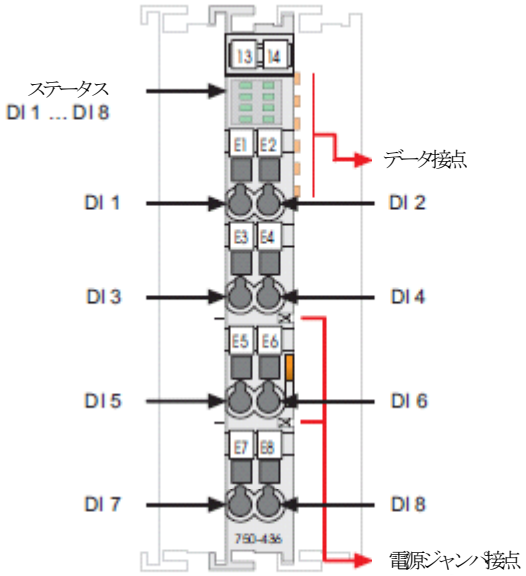
技術データ

型番　750-	432	433
入力点数	4	
消費電流（内部）	5.5mA	
供給電圧（電源ジャンパ接点経由）	DC 24V（-25%/+30%）	
信号電圧（0）	DC -3V～+5V	
信号電圧（1）	DC 15V～30V	
入力フィルター時定数	3ms	0.2ms
入力電流	4.5mA（平均）	
絶縁耐圧	500V（システム－電源間）	
内部ビット幅	4 ビット（入力）	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm²	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	

プロセスイメージ

入力ビット	B3	B2	B1	B0
意味	DI 4 信号状態 – チャンネル 4	DI 3 信号状態 – チャンネル 3	DI 2 信号状態 – チャンネル 2	DI 1 信号状態 – チャンネル 1

7.4　8 チャンネル デジタル入力（DC 24V／3.0ms、負方向スイッチ）
型番 750-436



技術説明

750-436 デジタル入力モジュールは、センサやスイッチなどのフィールド機器からの制御用信号の入力に用いられます。

このモジュールは単線信号接続用で 8 チャンネルの入力があります。従って、8 本のセンサ線を直接入力端子 DI 1～DI 8 に接続することができます。

本モジュールは負方向スッチング用です。つまり、入力チャンネルに 0V の電圧がかかった場合、その入力の状態は"high"になります。

各入力チャンネルには時定数が 3ms の RC ノイズ除去フィルタが付いています。
全ての入力チャンネルは未接続のときは電位が浮いています。
内部回路にはオプトカプラがあり、内部バスとフィールド側との間を電気的に絶縁するのに用いられています。

フィールドバス内ノードを設計するとき、入力モジュールはどんな位置でもコンフィグレーションできます。モジュールのタイプ毎にグループ化する必要はありません。

メモ
このモジュールは、後続のモジュールにフィールド機器用の電源ラインを通過させるために電源ジャンパ接点を持っています。

入力モジュール用の 24V フィールド電源は、隣のモジュールまたは電源入力モジュールから取り出します。フィールド機器用電源は、各 I/O モジュールの電源ジャンパ接点を介して自動的に供給されます。

注意

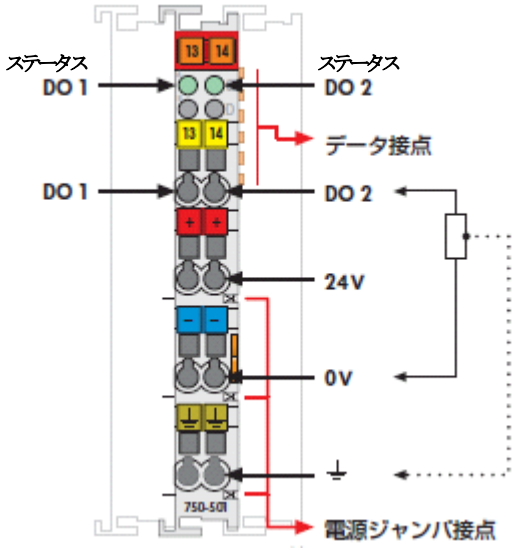
内部電源ジャンパ接点の最大電流容量は 10A です。システムを構築するときは、各モジュールの消費電流の合計が最大電流容量を超えないようにすることが重要です。もしこの許容量を超えるケースがあった場合は電源入力モジュールを追加しなければなりません。

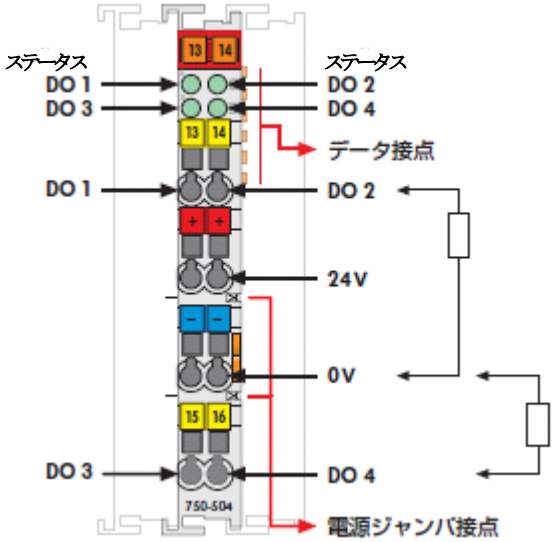
このモジュールはWAGO-I/O-SYSTEM の全てのバスカプラ／コントローラと共に使用することができます。

技術データ

型番 750-	436
入力数	8
消費電流（内部）	13mA（**最大**）
供給電圧（電源ジャンパ接点経由）	DC24V（-25%～+30%）
信号電圧（0）	DC -3V～+5V
信号電圧（1）	DC 15V～30V
入力フィルタ時定数	3ms
入力電流	2.8mA（**平均**）
絶縁耐圧	500V（**システム／電源間**）
内部ビット幅	8 **ビット（入力）**
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；**0.08～2.5mm^2
外形寸法（mm） W×H×L	12×64*×100（***キャリアーレールの上端から測定**）

7.5　デジタル出力（標準）
型番 750-501～504, 516, 519





技術説明

各々の動作電圧用には、電源モジュールを連結してK－Bus を接続することにより電源が供給されます。モジュールを DIN レール上に嵌め込むと、内部の電源ジャンパー接点を通して各モジュール間の電源接続が自動的に行われます。

２チャンネルデジタル出力の場合には、4 線式デバイスの接続（V+、0V、信号およびグラウンド）が標準となります。しかし 4 チャンネルデジタル出力モジュールの場合には、２線または３線デバイスを使用することになります。

ディジタル出力モジュールにはすべて短絡回路保護機能が用意されています。

過負荷状態から出力モジュールを保護する場合は必ずヒューズ付きの電源モジュール（750-601）を利用して電源を外部から供給してください。

モジュール 750-516 はネガティブ・スイッチング用(負方向接続用)のモジュールです。

次頁に表示された出力値は、100％のデューティサイクル条件下でのデータです。2A バージョンの場合には、より高い負荷電流で一つのチャンネルの動作を行うことが可能です。但しその際に、1 個のモジュール当たりの合計電流が 3.5A を超えないことを必ず確認してください。
例：1×3.0A、1×0.5A（合計電流：3.5A）

バス動作のビット数値割当ては、バスカップラに近いほうから始まって左から右への順番指定となっています。ノード／ステーション内部に取付ける異なる I/O モジュールの位置順を変えるとモジュールに対するビット割り当てが変わります。ただしアナログ出力モジュールや特殊モジュールが同一ノード内に存在する場合は、これらアナログ・特殊モジュールがすべてのディジタル出力モジュールに優先してビットが割り当てられます。（詳しくは第４章をご覧ください。）

技術データ

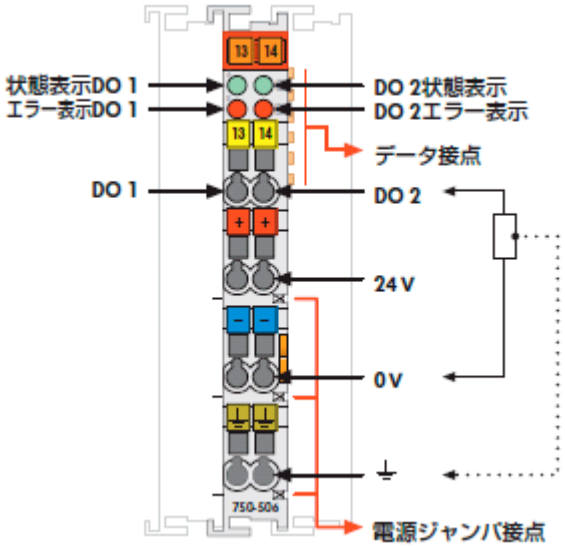
型番　750-	501	502
出力数	2	
負荷のタイプ	抵抗性、誘導性、ランプ	
定格電圧	24V DC（-15%/+20%）	
出力電流（DC）	0.5A	2A
消費電流（内部）	7mA	
絶縁耐圧	500V（システムー電源　間）	
内部ビット幅	2	
	短絡保護回路付	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm²	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	

型番　750-	504	516*)
出力数	4	
負荷のタイプ	抵抗性、誘導性、ランプ	
定格電圧	24V DC（-15%/+20%）	
出力電流（DC）	0.5A	
消費電流（内部）	15mA	
絶縁耐圧	500V（システムー電源　間）	
内部ビット幅	4	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm²	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	

*）ネガティブスイッチング（負方向接続）

型番　750-	519
出力数	4
負荷のタイプ	抵抗性、誘導性、ランプ
定格電圧	5V DC
出力電流（DC）	20mA
消費電流（内部）	16mA
絶縁耐圧	500V（システムー電源　間）
内部ビット幅	4
電線むき長さ	8-9mm
動作温度範囲	0°C～+55°C
電線接続	ケージクランプ；0.08～2.5mm²
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）

7.6　デジタル出力（診断機能付きの標準タイプ）
型番 750-506



技術説明

動作電圧用として、電源モジュールを連結して接続することにより電源が供給されます。モジュールを DIN レール上に嵌め込むと、内部の電源ジャンパー接点を通して各モジュール間の電源接続が自動的に行われます。

デジタル出力（750-506）は入力と出力の４ビットずつの内部ビット幅を持っています。出力ビットは４ビットのうち上位２ビットを使用して出力を制御します。このとき出力の短絡またはオープン回路（断線）の状況が入力の４ビットにて判断することが可能です。尚、診断機能付きのデジタル出力用として、4 線デバイス（V+、0V、信号およびグラウンド）が標準となります。デジタル出力にはすべて短絡回路保護機能が用意されています。

過負荷状態が予測される場合には、出力モジュールを保護するために必ずヒューズ付きの電源モジュール（750-601）
を電源と結線してください。

技術データ

型番　750-	506
出力数	2
消費電流（内部）	15mA
定格電圧	24V DC（-15%/+20%）
負荷のタイプ	**抵抗性、誘導性、ランプ**
出力電流（DC）	0.5A
診断	**オープン回路、過負荷**
絶縁耐圧	500V（**システム－電源　間**）
内部ビット幅	**4 入力、4 出力**
動作温度範囲	0°C～+55°C
電線接線	**ゲージクランプ；**0.08～2.5mm²
電線むき長さ	8-9mm
外形寸法（mm）　W×H×L	12×64*×100（***キャリアレールの上端から測定**）

出力ビットによって、デジタル出力の状態を制御します。

＜出力４ビットの内容＞

	Bit 3	Bit 2	Bit 1	Bit 0
機能	**機能なし**	**機能なし**	**O2 出力の制御**	**O1 出力の制御**

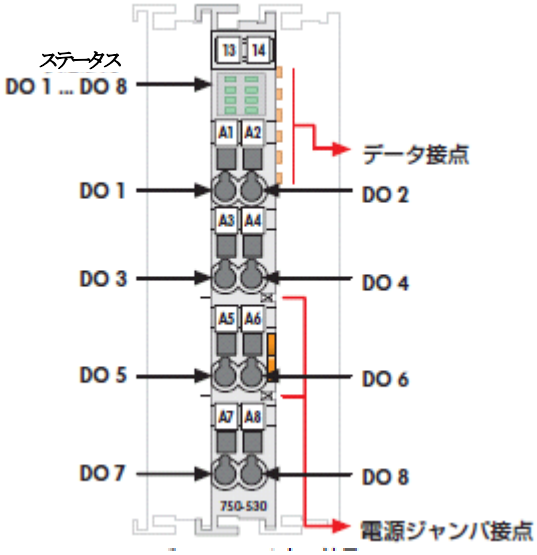
入力ビットによって、デジタル出力の状態を診断できます。

＜入力４ビットの内容＞

	Bit 3	Bit 2	Bit 1	Bit 0
機能	**O2 出力の診断**		**O1 出力の診断**	
出力は出力ビットに従う（正常）	0	0	0	0
負荷が接続されていない（断線、またはオープン）	0	1	0	1
短絡（出力回路のショート）	1	0	1	0
電源電圧が極端に低い*	1	1	1	1

* 診断ビットはヒステリシスの動きをします。すなわち、フィールド側の電圧が立ち下がりサイクルで 11V よりも高ければ、診断ビットはオンに切替えられます。また、立ち上がりサイクルで電圧が 15.5V よりも低ければ、診断ビットはオフに切替えられます。

7.7　8 チャンネル デジタル出力（DC24V／0.5A、正方向スイッチ）
型番 750-530



技術説明

各 I/O モジュールにはそれぞれの動作電圧用として、電源の供給は連続に接続された端子部を通して行われます。I/O モジュールを DIN レール上に嵌め込むと、各モジュール間で電源接続が自動的に行われます。

注意：
モジュール 750-530 はポジティブ・スイッチング（正方向接続）用のデジタル出力モジュールです。
各々の端子は出力信号が ON のとき＋側（＋24V）の信号が出力されます。

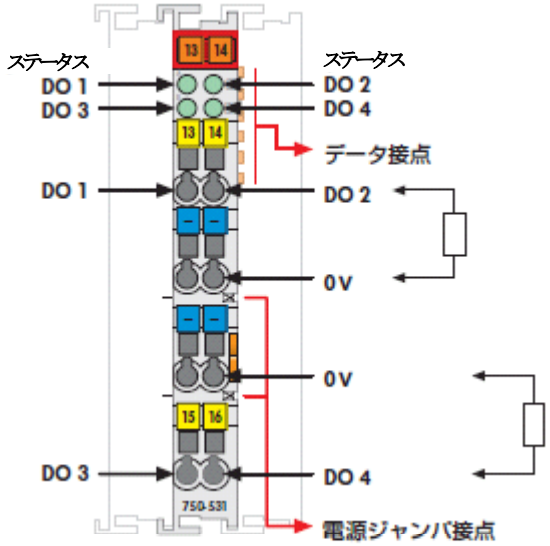
配線時には、I/O システムに供給する DC 0V のコモン電源－側の端子を外部で準備してください。

バス動作におけるビット数値割当ては、バスカップラに近い方から始まって左から右へ順番に指定されます。ノード／ステーション内部に取付ける種々の異なる I/O モジュールの位置は順番を変えることによってビット割り当てが決まります。ただしアナログ出力モジュールや特殊モジュールが存在する場合には、これらのアナログ・特殊モジュールは優先的にビット割り当てが行われます。

技術データ

型番 750-	530
出力数	8
負荷のタイプ	**抵抗性、誘導性、ランプ**
定格電圧	24V DC（-15%/+20%）
出力電流（DC）	0.5A
消費電流（内部）	25mA
絶縁耐圧	500V **システム／電源**
内部ビット幅	8
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；**0.08～2.5mm²
電線むき長さ	8-9mm
外形寸法（mm） W×H×L	12×64*×100（***キャリアレールの上端から測定**）

7.8　4 チャンネル デジタル出力（DC 24V／0.5A、2 線式）
型番 750-531



技術説明

本モジュールに接続された出力機器は、制御システムからモジュールに送られたデジタル出力信号により駆動されます。
このモジュールは 4 点の出力チャンネルを持っており、4 個の 2 線式アクチュエータを直接モジュールに接続することができます。
アクチュエータは各出力端子 DO 1、DO 2、DO 3、DO 4 と 0V 端子間にそれぞれ接続できます。
3 線式アクチュエータをこのモジュールにつなぐ場合は、24V 電源の供給用に電源端子拡張モジュール（750-603 または 750-614）を用意する必要があります。

注意
誘導性負荷を接続する場合は、リカバリダイオードなどの保護回路をその負荷と並列につなぐ必要があります。

出力チャンネルは短絡保護回路を備えております。また、正方向スイッチ出力になります。
これは出力チャンネルの電圧がフィールド機器電源の 24V 側にスイッチしたとき、出力チャンネルの状態は "high"になることを意味します。

4 点の出力チャンネルの状態は緑色のステータス LED によって示されます。
内部バスとフィールド機器間の電気的な絶縁のために、オプトカプラが使用されています。

フィールドバス内ノードを設計するとき、出力モジュールはどんな位置でもコンフィグレーションできます。
モジュールのタイプ毎にグループ化する必要はありません。

出力モジュール用の 24V フィールド電源は、隣のモジュールまたは電源入力モジュールから取り出します。
フィールド機器用電源は、各 I/O モジュールの電源ジャンパ接点を介して自動的に供給されます。

注意

内部電源ジャンパ接点の最大電流容量は 10A です。システムを構築するときは、各モジュールの消費電流の合計が最大電流容量を超えないようにすることが重要です。もしこの許容量を超えるケースがあった場合は電源入力モジュールを追加しなければなりません。

注意

過負荷の場合、ヒューズ付き電源入力モジュール（750-601）を出力モジュールの保護のために電源ライン側に接続しなければなりません。。

この出力モジュールは、WAGO-I/O-SYSTEM の全てのバスカプラ／コントローラと共に使用することができます。

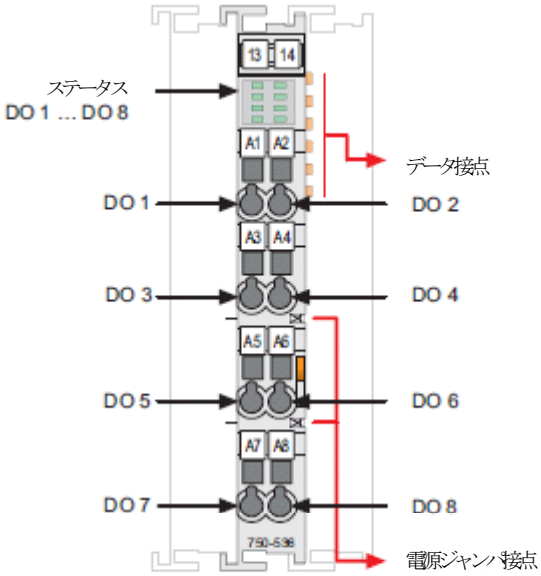
技術データ

型番 750-	531
出力数	4
消費電流（内部）	7mA
供給電圧（電源ジャンパ接点経由）	DC24V（-25%～+30%）
負荷のタイプ	**抵抗性、誘導性、ランプ**
スイッチ速度	**最大** 1kHz
逆電圧保護回路	**あり**
出力電流（DC）	0.5A（**短絡保護付き**）
エネルギ消費 W_{max}（誘導性負荷 OFF 時）	0.3J $L_{max}=2\ W_{max}/I^2$
消費電流（フィールド側）	**平均** 30mA／**モジュール＋負荷電流**
絶縁耐圧	500V（**システム／電源間**）
内部ビット幅	4 **ビット（出力）**
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；** 0.08～2.5mm²
電線むき長さ	8-9mm
外形寸法（mm） W×H×L	12×64*×100（***キャリアレールの上端から測定**）

プロセスイメージ

出力ビット	B3	B2	B1	B0
意味	DO 4 制御信号 – チャンネル 4	DO 3 制御信号 – チャンネル 3	DO 2 制御信号 – チャンネル 2	DO 1 制御信号 – チャンネル 1

7.9　8 チャンネル デジタル出力（DC 24V／0.5A、負方向スイッチ）
型番 750-536



技術説明

本モジュールに接続された出力機器は、制御システムからモジュールに送られたデジタル出力信号により駆動されます。
このモジュールは 8 点の出力チャンネルを持っており、8 個のアクチュエータを直接各出力端子 DO 1～DO 8 にそれぞれ接続できます。

注意
誘導性負荷を接続する場合は、リカバリダイオードなどの保護回路をその負荷と並列につなぐ必要があります。

出力チャンネルは短絡保護回路を備えております。また、負方向スイッチ出力になります。
これは出力チャンネルの電圧がフィールド機器電源の 0V 側にスイッチしたとき、出力チャンネルの状態は "high"になることを意味します。

8 点の出力チャンネルの状態は緑色のステータス LED によって示されます。
内部バスとフィールド機器間の電気的な絶縁のために、オプトカプラが使用されています。

フィールドバス内ノードを設計するとき、出力モジュールはどんな位置でもコンフィグレーションできます。モジュールのタイプ毎にグループ化する必要はありません。

出力モジュール用の 24V フィールド電源は、隣のモジュールまたは電源入力モジュールから取り出します。フィールド機器用電源は、各 I/O モジュールの電源ジャンパ接点を介して自動的に供給されます。

注意
内部電源ジャンパ接点の最大電流容量は 10A です。システムを構築するときは、各モジュールの消費電流の合計が最大電流容量を超えないようにすることが重要です。もしこの許容量を超えるケースがあった場合は電源入力モジュールを追加しなければなりません。

注意

過負荷の場合、ヒューズ付き電源入力モジュール（750-601）を出力モジュールの保護のために電源ライン側に接続しなければなりません。。

この出力モジュールは、WAGO-I/O-SYSTEM の全てのバスカプラ／コントローラと共に使用することができます。

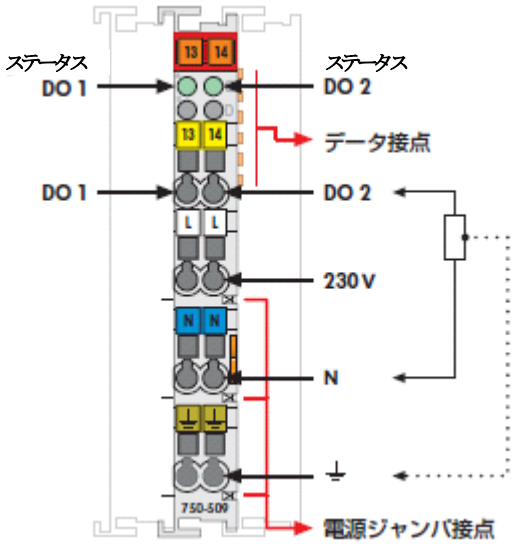
技術データ

型番 750-	536
出力数	8
消費電流（内部）	25mA
供給電圧（電源ジャンパ接点経由）	DC24V（-25%～+30%）
負荷のタイプ	**抵抗性、誘導性、ランプ**
スイッチ速度	**最大** 2kHz
逆電圧保護回路	**あり**
出力電流（DC）	0.5A（**短絡保護付き**）
エネルギ消費 W_{max}（誘導性負荷 OFF 時）	0.5J $L_{max}=2\ W_{max}/I^2$
消費電流（フィールド側）	**平均** 12mA／**モジュール＋負荷電流**
絶縁耐圧	500V（**システム／電源間**）
内部ビット幅	8 **ビット（出力）**
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；** 0.08～2.5mm²
電線むき長さ	8-9mm
外形寸法（mm） W×H×L	12×64*×100（***キャリアレールの上端から測定**）

プロセスイメージ

出力ﾋﾞｯﾄ	B7	B6	B5	B4	B3	B2	B1	B0
意味	DO 8 制御信号－ﾁｬﾝﾈﾙ 8	DO 7 制御信号－ﾁｬﾝﾈﾙ 7	DO 6 制御信号－ﾁｬﾝﾈﾙ 6	DO 5 制御信号－ﾁｬﾝﾈﾙ 5	DO 4 制御信号－ﾁｬﾝﾈﾙ 4	DO 3 制御信号－ﾁｬﾝﾈﾙ 3	DO 2 制御信号－ﾁｬﾝﾈﾙ 2	DO 1 制御信号－ﾁｬﾝﾈﾙ 1

7.10 デジタル出力（ソリッドステートリレー）
型番 750-509



技術説明

ソリッドステートリレー・モジュール用の電源は、230V の動作電圧用の電源入力モジュールを連結して接続することにより供給されます。モジュールを DIN レール上に嵌め込むと、内部の電源ジャンパー接点を通して各モジュール間の電源接続が自動的に行われます。

注意

制御側の電源は、電源ジャンパー接点を通さずに電子回路部から直接供給されます。したがって、スイッチング素子の各出力接点は常にフィールド側に取付けられています。これらの接点において片側の端子部分を電源に直接結線しなければなりません。

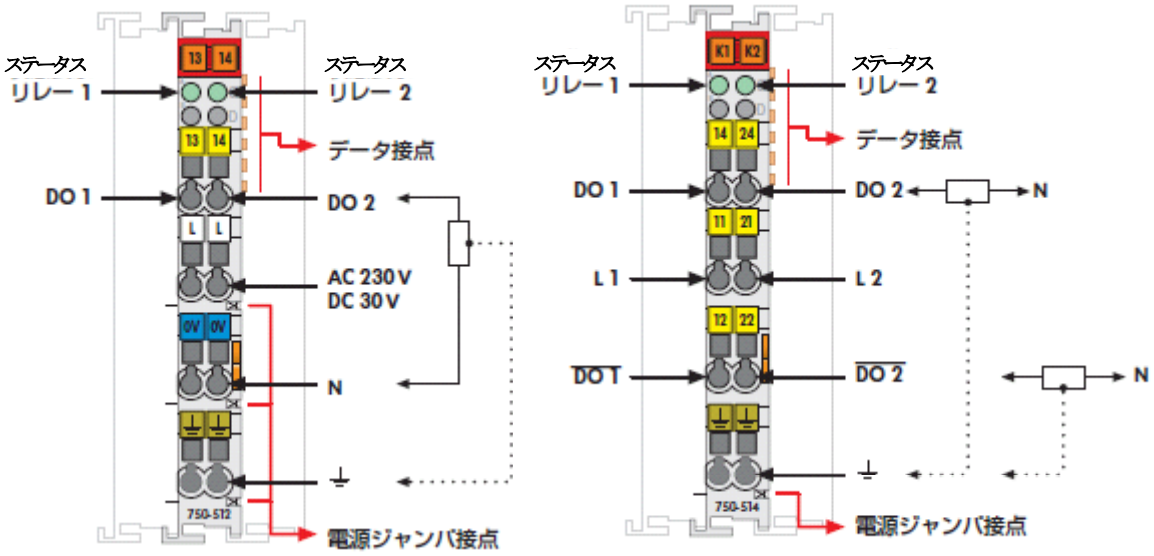
デジタル出力にはすべて短絡回路保護機能が用意されています。**過負荷状態の場合に対処して、出力モジュールを保護するために必ずヒューズ付きの電源入力モジュール（750-609）を電源と結線してください。**

バス動作の標準的なビット数値割当ては、バスカプラに近い方から始まって左から右への順番指定となっています。構成したステーション内部に取付ける各 I/O モジュールの位置は、ユーザが任意に選択できます。WAGO-I/O-SYSTEM のすべてのバスカプラに対し、この出力モジュールを接続することが可能です。

技術データ

型番　750-	509
出力数	2
消費電流（内部）	10mA
スイッチング電圧	0V～230V AC/DC
スイッチ電流	300mA AC（最大）
動作速度	1.65ms（平均）、5ms（最大）
オン時抵抗値	2.1Ω（平均）3.2Ω（最大）
インパルス電流	0.5A（20s）、1.5A（0.1s）
過電圧保護	＞±380V（バリスター）
絶縁耐圧	1.5kV システム／電源
内部ビット幅	2
動作温度範囲	0°C～+55°C
電線接線	ゲージクランプ；0.08～2.5mm^2
電線むき長さ	8-9mm
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）

7.11 デジタル出力（リレー）
型番 750-512...514, 517



技術説明

リレーコイル用の電源は電源ジャンパー接点を通さずに、電子回路部から直接供給されます。一方、リレーの各出力接点は常にフィールド側に置かれます。

注意

バージョン 1：ノンフローティング（モジュール 750-512）

動作電圧用としては、連結接続した電源端子を通して電源が供給されます。モジュールを DIN レール上に嵌め込むと、各モジュール間の電源接続が自動的に行われます。これらの接点の中から 1 つの端子部分を電源に直接結線しなければなりません。

バージョン 2：分離出力（モジュール 750-513、750-514）

これらの I/O モジュールには電源ジャンパー接点が用意されていません。このモジュールに電源結線を行う際には、別途電源を用意してください。

バス動作の標準的なビット数値割当ては、バスカプラに近い方から始まって左から右への順番指定となっています。構成したステーション内部に取付ける各 I/O モジュールの位置はユーザが任意に選択できます。WAGO-I/O-SYSTEM のすべてのバスカプラに対し、この出力モジュールを接続することが可能です。

技術データ

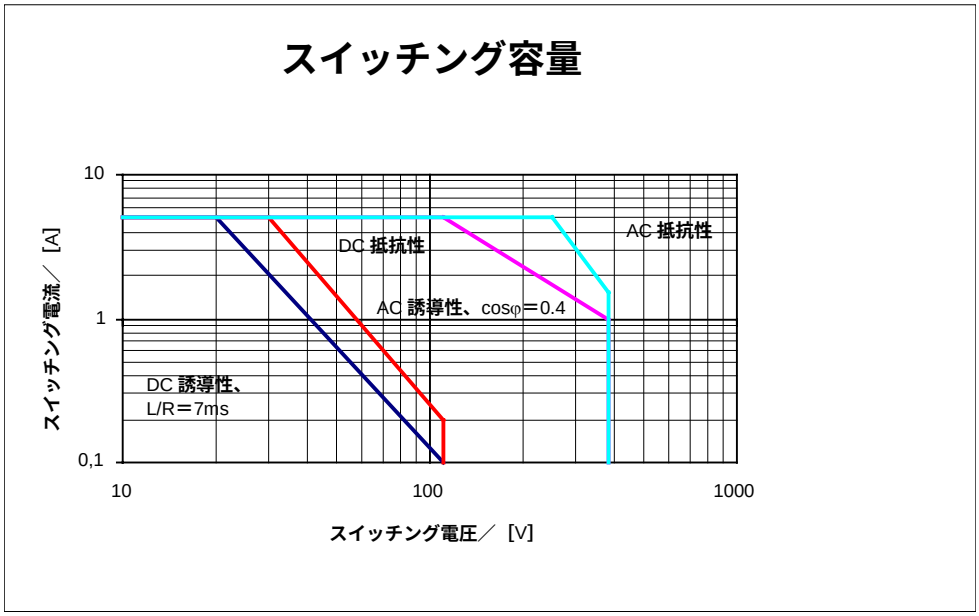
型番　750-	512	513
接点のタイプ	**2 個のメーク接点**	
消費電流（内部）	100mA（**最大値**）	
スイッチング電圧	30V DC；250V AC	
スイッチング電力	60W*；500VA $\cos\rho_{max}$ ＝0, 4, L/R_{max}＝7ms	
スイッチング電流	2A AC/DC*	
絶縁耐圧	4kV **システム／電源**	
内部ビット幅	2	
動作温度範囲	0°C～+55°C	
電線接続	**ケージクランプ**；0.08～2.5mm^2	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（***キャリアレールの上端から測定**）	

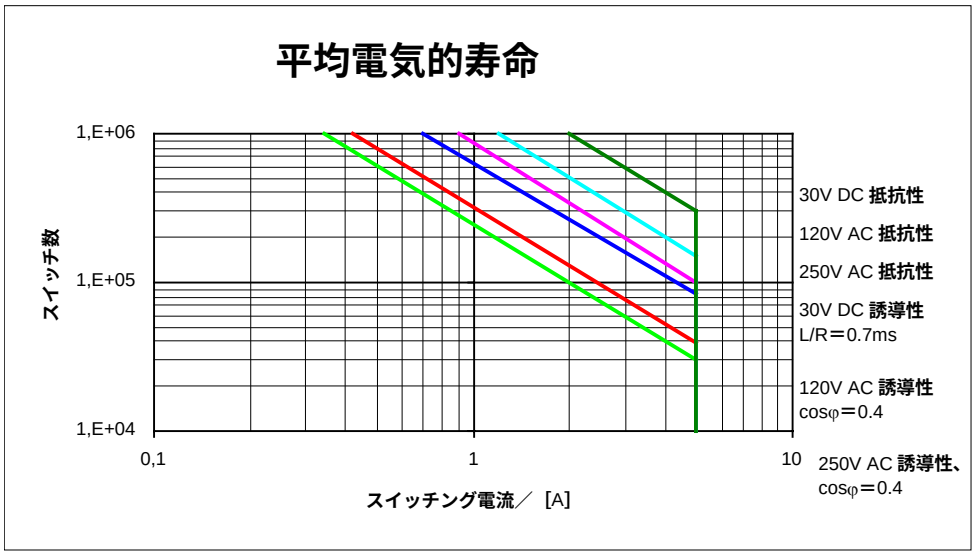
型番　750-	514	517
接点のタイプ	**2 個の切替え接点**	
消費電流（内部）	70mA（**最大値**）	105mA（**最大値**）
スイッチング電圧	30V DC；125V AC	250V AC；300VDC
スイッチング電源	30W；62.5VA	400VA
スイッチング電流	0.5A AC/1A DC	1A AC
絶縁耐圧	1.5kV／**電源**	4kV **システム／電源**
内部ビット幅	2	
動作温度範囲	0°C～+55°C	
電線接続	**ケージクランプ**；0.08～2.5mm^2	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（***キャリアレールの上端から測定**）	

注意

実際の運用では本リレーモジュールで大きな逆起電力を発生する可能性のある誘導性負荷を直接駆動したり、規定された寿命近くまで開閉を繰り返すことは避けるようにしてください。誘導性負荷を接続する場合の注意については、後述の「**リレーモジュール用接点保護回路**」の項を参照ください。

モジュール 750-512 と 750-513 に使用されているリレー：

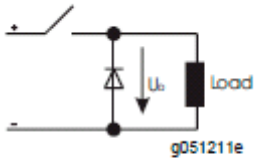
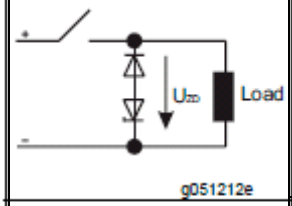
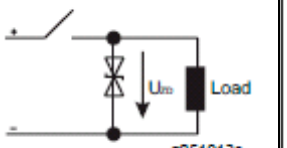
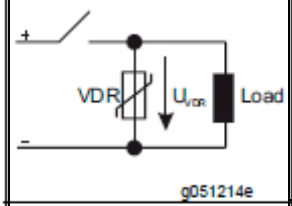
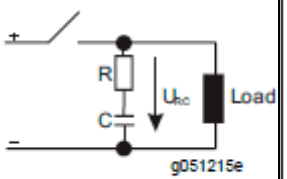




リレーモジュール用接点保護回路

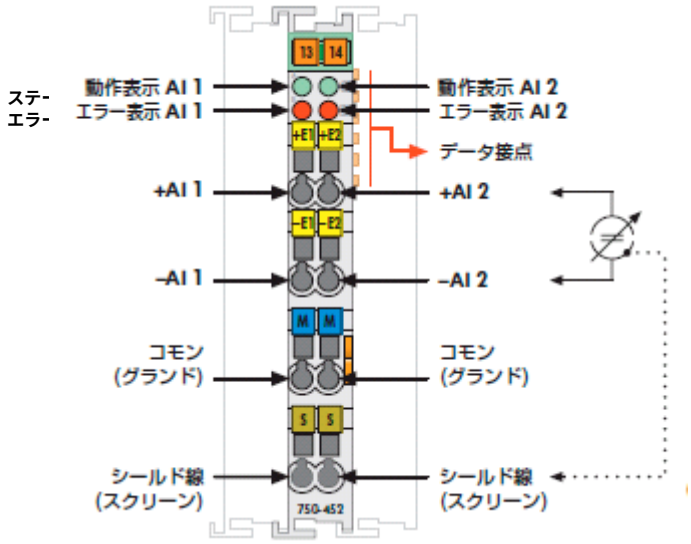
リレーモジュールに誘導性負荷（コンタクタ、ソレノイドバルブなど）をつないでスイッチング動作をさせる場合、回路を OFF にしたとき接点間に数千 V に至る逆起電圧が発生する可能性があります。実際には、この逆起電圧が EMC 基準に定められた許容値を超えることが非常に多くあります。従って、これを抑える為には外部に逆起電圧抑制回路を設けなければなりません。更に、負荷の電流が比較的少ない場合（1A 以下）でも、逆起電圧がアーク放電を起こし、接点を破壊することがあります。その場合リレーの寿命や安全性に重大な影響を与えることになります。

以下の保護回路は誘導性負荷を使用するユーザが負荷側に並列に、直接接続すべきものですが、実際のシステムで必ず確認試験をしておく必要があります。

負荷保護回路	OFF 時自加遅延時間	誘導電圧制限の指定	AC 電圧適用	利点／欠点
Diode 	長い	あり（U_D）	なし	利点 —使用が簡単 —低コスト、高信頼性 —寸法問わず —低誘導電圧 欠点 —負荷抵抗分で抑制 —OFF 時の遅れが長い
Series connection diode/zener diode 	中位から短い	あり（U_{ZD}）	なし	利点 —寸法問わず 欠点 U_{ZD}（ツェナー電圧）以上でのみ抑制
Suppressor diode 	中位から短い	あり（U_{ZD}）	あり	利点 —低コスト —寸法問わず —正負両方向ピーク電圧の制限 —AC 電圧に適用可 欠点 —U_{ZD}以上でのみ抑制
Varistor 	中位から短い	あり（U_{VDR}）	あり	利点 —高電圧を吸収 —寸法問わず —AC 電圧に適用可 欠点 —U_{VDR}（バリスタ電圧）以上でのみ抑制
R/C combination 	中位から短い	なし	あり	利点 —電力保存により高周波抑制 —AC 電圧に適用可 —電圧レベルに関係なく抑制 欠点 —正確な寸法が必要 —進入電流が大きい

8　アナログ入力モジュール

8.1　2 チャンネルアナログ入力 0～20mA/4～20mA（外部電源利用型）
型番 750-452, 454



技術説明

入力チャンネルは差動入力で、外部の電源を必要とします。コモングラウンド端子が用意されています。入力は+I と-I の各端子を通して結線します。シールド端子は DIN レールに結線されます。モジュールを DIN レール上に嵌め込むと、各モジュール間のシールド接続が自動的に行われます。

これらの I/O モジュールには電源ジャンパー接点が取付けられていません。電源の供給は、外部で準備してください。

注意：
このモジュール電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュールはこのモジュールの右側に接続することはできません。例えば 3 個の電源ジャンパー接点が用意されたモジュールをこのモジュールの右側に追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	452	454
チャンネル数	2	
駆動電圧	システム電圧経由	
消費電流（内部）	70mA	
最大許容電圧	35V（最大）	
信号電流	0〜20mA	4〜20mA
入力インピーダンス	50Ω（平均）	
分解能	12Bit	
絶縁耐圧	500V（システムー電源　間）	
変換時間	2ms（平均）	
内部ビット幅	2ch×16 ビット	
動作温度範囲	0°C〜+55°C	
測定精度	フルスケール±０．２%以内	
電線接続	ケージクランプ；0.08〜2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）	
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7	
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)	

数値フォーマット

アナログ値はすべて数値単位フォーマットで表示されます。分解能は 13 ビットです。使用される数値フォーマットを下記の表に示します。3 個の最下位ビットについては考慮していません。

入力電流 0〜20mA	入力電流 4〜20mA	バイナリ値	16 進数値	10 進数値
20	20	0111 1111 1111 1000	7F F8	32760
10	12	0100 0000 0000 0000	40 00	16384
5	8	0010 0000 0000 0000	20 00	8192
2.5	6	0001 0000 0000 0000	10 00	4096
0.156	4.125	0000 0001 0000 0000	01 00	256
0.01	4.0078	0000 0000 0001 0000	00 10	16
0.005	4.0039	0000 0000 0000 1000	00 08	8
0	4	0000 0000 0000 0111	00 07	7
0	4	0000 0000 0000 0000	0	0

バイナリデータと入力電流値の関係

入力電流値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

① 750-452 の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0 （16 進数）

↓

1024 （10 進数）

$\therefore (1024 / 2^{12}) \times 20\text{mA} = 5\text{mA}$

② 750-454 の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

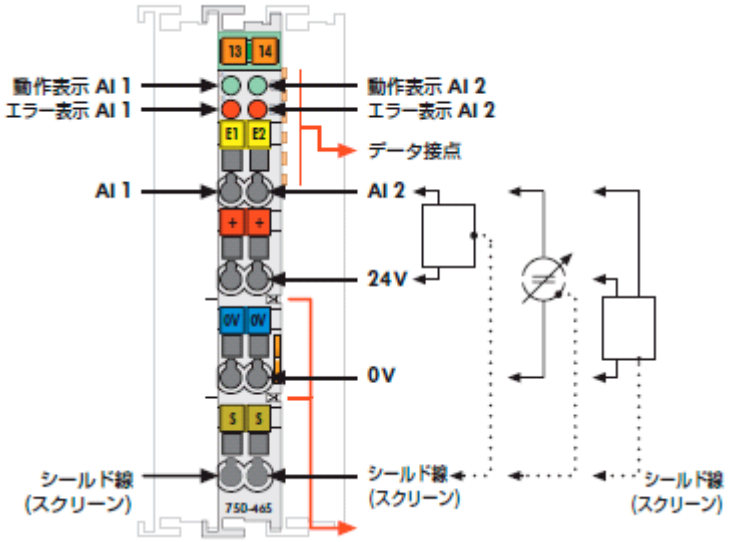
4 0 0 （16 進数）

↓

1024 （10 進数）

$\therefore 4\text{mA} + (1024 / 2^{12}) \times (20\text{mA} - 4\text{mA}) = 8\text{mA}$

8.2　2 チャンネルアナログ入力 0～20mA/4～20mA
（シングルエンド入力：システム電源供給型）
型番 750-465（12 ﾋﾞｯﾄ 0～20mA）　750-472（15 ﾋﾞｯﾄ 0～20mA）
750-466（12 ﾋﾞｯﾄ 4～20mA）　750-474（15 ﾋﾞｯﾄ 4～20mA）



技術説明

これらのモジュールは入力電流を発生させるための電源 24VDC 電源ジャンパーを通して得られるセンサー用電源を利用します。入力は+I の端子に結線します。24V/0V の各端子を通して、センサをこのモジュールに直接結線することが可能です。

入力チャンネルはシングルエンド入力で、コモングラウンド電位が使用されています。

シールドは DIN レールに結線されます。モジュールを DIN レール上に嵌め込むと、各モジュール間のシールド結線が自動的に行われます。

注意：
これらのモジュールには一番下の部分の電源ジャンパー接点が用意されていません。したがって、3 個全部の電源ジャンパー接点が必要なモジュール（例：2 チャンネルのデジタル入力モジュール）は、これらのモジュールの右側に接続することができません。

技術データ

型番　750-	465	472	466	474
入力数	2			
外部への駆動電圧	電源ジャンパー接点を通して 24V DC（-15%/+20%）			
消費電流（内部）	75mA			
最大定格電圧	29V（最大）			
最大入力電流	24V 38mA 以下（１分間）			
信号電流	0～20mA		4～20mA	
入力インピーダンス	150Ω以下	50Ω	150Ω以下	50Ω
分解能	12 ビット	15 ビット	12 ビット	15 ビット
絶縁耐圧	500V（システムー電源　間）			
変換時間	2ms	80ms	2ms	80ms
内部ビット幅	2×16 ビット			
動作温度範囲	0°C～+55°C			
保存温度範囲	－25°C～+85°C			
測定精度	FS±０.2%以内	FS±０.1%以内	FS±０.2%以内	FS±０.1%以内
電線接続	ケージクランプ；0.08～2.5mm²			
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）			
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7			
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)			

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。分解能は 12 ビットです。使用される数値フォーマットを下記の表に示します

①７５０－４６５、７５０－４６６の場合

入力電流 (0～20mA)	入力電流 (4～20mA)	バイナリ値	16 進数値	10 進数値
20	20	0111 1111 1111 1000	7F FF	32767
10	12	0100 0000 0000 0000	40 00	16384
5	8	0010 0000 0000 0000	20 00	8192
2.5	6	0001 0000 0000 0000	10 00	4096
0.156	4.125	0000 0001 0000 0000	01 00	256
0.01	4.0078	0000 0000 0001 0000	00 10	16
0.005	4.0039	0000 0000 0000 1000	00 08	8
0	4	0000 0000 0000 0111	00 07	7
0	4	0000 0000 0000 0000	0	0

②７５０－４７２、７５０－４７４の場合

入力電流 (0～20mA)	入力電流 (4～20mA)	バイナリ値	16 進数値	10 進数値
20	20	0111 1111 1111 1111	7F FF	32767
10	12	0100 0000 0000 0000	40 00	16384
5	8	0010 0000 0000 0000	20 00	8192
2.5	6	0001 0000 0000 0000	10 00	4096
0.156	4.125	0000 0001 0000 0000	01 00	256
0.097	4.0078	0000 0000 0001 0000	00 10	16
0.0048	4.0039	0000 0000 0000 1000	00 08	8
0.0042	4.0034	0000 0000 0000 0111	00 07	7
0	4	0000 0000 0000 0000	0	0

バイナリーデータから入力電流値を計算する。

① 750-465 の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0 （16 進数）

↓

1024 （10 進数）

∴ $(1024 / 2^{12}) \times 20\text{mA} = 5\text{mA}$

② 750-466 の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0 （16 進数）

↓

1024 （10 進数）

∴ $4\text{mA} + (1024 / 2^{12}) \times (20\text{mA} - 4\text{mA}) = 8\text{mA}$

③ 750-472 の場合

0 010 0000 0000 0000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この１５ﾋﾞｯﾄを使います。

010 0000 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

2 0 0 0 （16 進数）

↓

8192 （10 進数）

∴ $(8192 / 2^{15}) \times 20\text{mA} = 5\text{mA}$

④ **750-474 の場合**

0 010 0000 0000 0000　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この１５ﾋﾞｯﾄを使います。

010 0000 0000 0000 0000　（１５ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

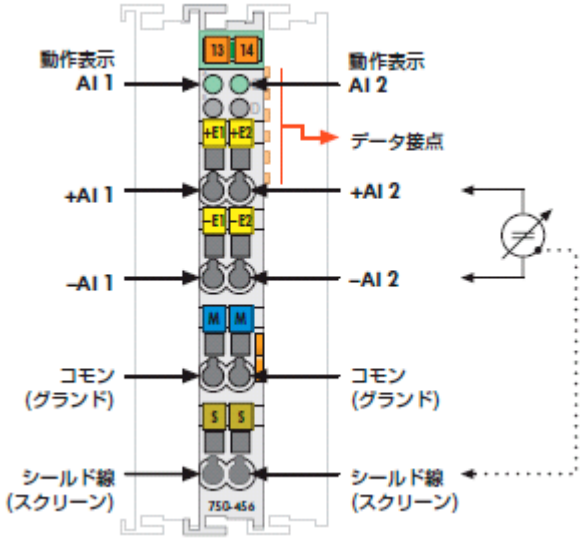
2 0 0 0　（16 進数）

↓

8192　（10 進数）

∴ **4mA ＋（8192 / 2^{15}）×（20mA－4mA）＝8mA**

8.3　2 チャンネルアナログ入力 +/-10V（差動、外部電源利用型）型番 750-456（12 ﾋﾞｯﾄ）



技術説明

入力チャンネルは差動入力で、コモングラウンド電位が用意されています。電源については外部での駆動電源が必要となります。

入力は+I と-I の各端子を通して結線します。シールドは DIN レールに結線されます。すなわちモジュールを DIN レール上に嵌め込むと、各モジュール間のシールド接続が自動的に行われます。

このモジュールには電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュールは、このモジュールの右側に直接接続することはできません。３個の電源ジャンパー接点が用意されたモジュールを追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	456
チャンネル数	2
駆動電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	65mA
最大定格電圧	35V（最大）
入力信号電圧	±10V
入力インピーダンス	570kΩ
分解能	12Bit
絶縁耐圧	500V（システムー電源間）
変換時間	2ms
内部ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
保存温度範囲	-25°C～+85°C
測定精度	±0.2%F.S.以内
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)

数値フォーマット

７５０－４５６

入力電圧±10V	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32760
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0	0000 0000 0000 0111	0	00
0	0000 0000 0000 0000	0	00
-2.5	1110 0000 0000 0000	E0 00	57344
-5	1100 0000 0000 0000	C0 00	49152
-7.5	1010 0000 0000 0000	A0 00	40960
-10	1000 0000 0000 0000	80 00	32768

バイナリーデータから入力電圧値を計算する。

入力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

① データが＋側の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

先頭は必ず「0」

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0 （16 進数）

↓

1024 （10 進数）

$\therefore (1024 / 2^{12}) \times 10\mathrm{V} = 2.5\mathrm{V}$

② データが－側の場合

先頭は必ず「1」です。

1110 0000 0000 0000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓データを反転します。

0 001 1111 1111 1 111

↓この 12 ﾋﾞｯﾄを使います。

0011 1111 1111 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

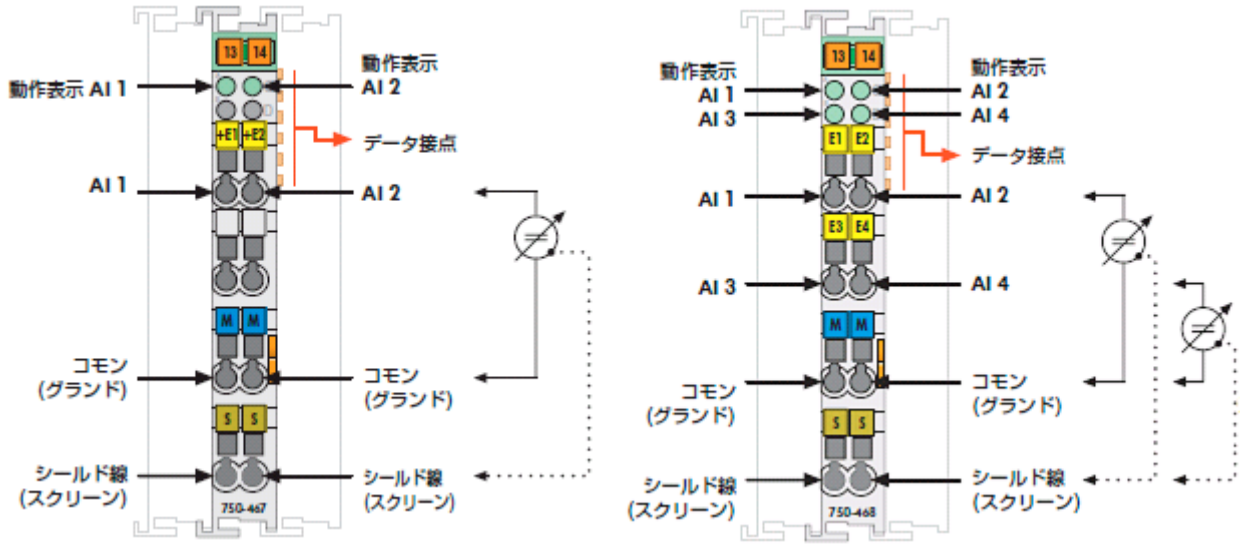
3 F F （16 進数）

↓

1023 （10 進数）

$\therefore -10 \times (1023 + 1) / 2^{12} = -1024 / 4096 = -2.5\mathrm{V}$

8.4　2、4 チャンネルアナログ入力 0～10V（シングルエンド入力）
型番 750-467, 468



技術説明

モジュール 750-467／750-468 は 0～10V の電圧信号入力モジュールです。入力点数として 1 チャンネルあたり 16 ﾋﾞｯﾄを占有しますがこのうち 12 ﾋﾞｯﾄを用いて入力電圧を表現します。

入力チャンネルはシングルエンド入力で、コモングラウンド電位が使用されています。

入力は+I と C の各端子を通して結線します。シールドは DIN レールに結線されます。モジュールを DIN レール上に嵌め込むと、各モジュール間の電源結線が自動的に行われます。

これらの I/O モジュールには電源ジャンパー接点が用意されていません。電源の供給は、データ接点から DC-DC コンバータを通して行われます。この各モジュールは内部からの電源によって動作します。

注意：
これらのモジュールには電源ジャンパー接点が用意されていません。したがって、3 個全部の電源ジャンパー接点が必要なモジュール（例：2 チャンネルのデジタル入力モジュール）をこのモジュールの右側に追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	467	468
チャンネル数	2	4
定格電圧	システム電圧経由（DC-DC コンバータ）	
消費電流（内部）	60mA	
電圧	35V（最大）	
信号電圧	0～10V	
入力インピーダンス	133kΩ（平均）	
分解能	12Bit	
絶縁耐圧	500V（システムー電源　間）	
変換時間	2ms（平均）	
内部ビット幅	2ch×16 ビットデータ	
動作温度範囲	0°C～+55°C	
測定精度	フルスケール±０．３%以内	
電線接続	ケージクランプ；0.08～2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）	
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7	
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)	

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。分解能は 12 ビットです。使用される数値フォーマット例を下記の表に示します（モジュール 750-467，468）。3 個の最下位ビットについては考慮していません。

入力電圧（0～10V）	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32760
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 1000	00 08	8
0	0000 0000 0000 0111	00 07	7
0	0000 0000 0000 0000	0	0

バイナリーデータと入力電圧の関係

入力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

0 010　　0000　　0000　　0 000　　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100　　0000　　0000　　（12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4　　0　　0　　（16 進数）

↓

1024　　（10 進数）

$$\therefore\ (1024 / 2^{12})\ \times\ 10V\ =\ 2.5V$$

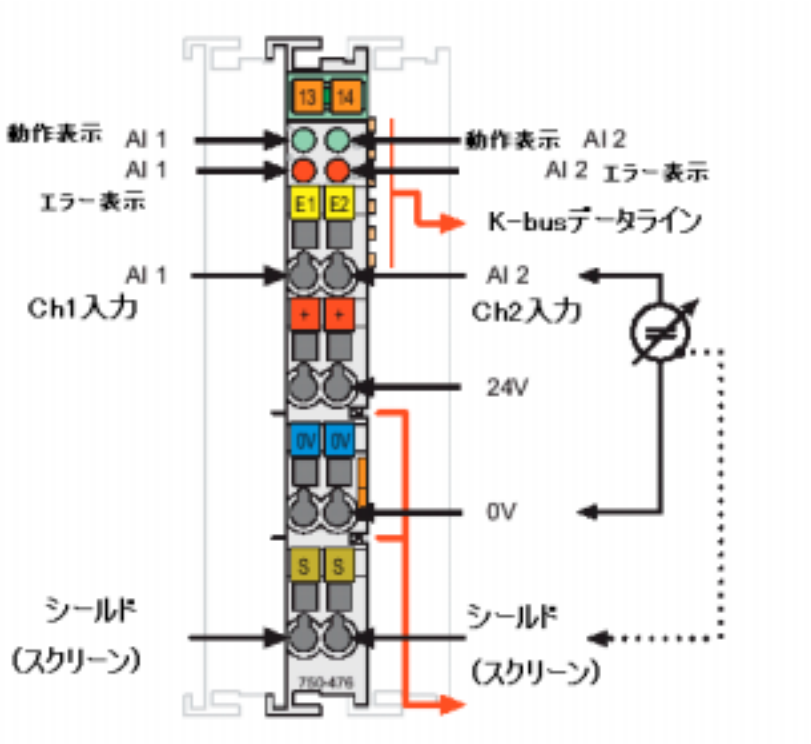
あるいは 12 ﾋﾞｯﾄバイナリデータから、

0100　0000　0000

2^{10}

$$\therefore\ (\ 2^{10} / 2^{12})\ \times\ 10V\ =\ 2.5V$$

8.5　2 チャンネルアナログ入力　–10V～+10V/0～10V（15 ﾋﾞｯﾄ高分解能型）
型番 750-476,478



技術説明

この２つのモジュールは分解能が 15 ビットで、各々2 点電圧入力モジュールです。

モジュール 750-476 は－10V～＋10V の電圧信号入力モジュールです。入力点数として１チャンネルあたり 16 ﾋﾞｯﾄを占有しますがこのうち１５ﾋﾞｯﾄで数値を用いて入力電圧を表現します。

モジュール 750-478 は０～10V の電圧信号入力モジュールです。入力点数として１チャンネルあたり 16 ﾋﾞｯﾄを占有しますがこのうち１５ﾋﾞｯﾄで数値を、先頭１ビットで符号を表現します。

入力チャンネルはシングルエンド入力で、コモングラウンド電位が使用されています。

入力は+I と C の各端子を通して結線します。シールドは DIN レールに結線されます。モジュールを DIN レール上に嵌め込むと、各モジュール間の電源結線が自動的に行われます。

これらの I/O モジュールには電源ジャンパー接点が用意されていません。電源の供給は、データ接点から DC-DC コンバータを通して行われます。この各モジュールは内部からの電源によって動作します。

技術データ

型番　750-	476	478
チャンネル数	2	
定格電圧	システム電圧経由（DC-DC コンバータ）	
消費電流（内部）	75mA	
電圧	35V（最大）	
信号電圧	-10V～+10V	0～10V
入力インピーダンス	130kΩ（平均）	
分解能	15Bit　およびサインﾋﾞｯﾄ	
絶縁耐圧	500V（システムー電源　間）	
変換時間	80ms（平均）	
内部ビット幅	2ch×16 ビットデータ	
動作温度範囲	0°C～+55°C	
測定精度	フルスケール±０．１%以内	
電線接続	ケージクランプ；0.08～2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）	
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7	
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)	

数値フォーマット

＜７５０－４７６　、　７５０－４７８＞

アナログ値はすべてバイナリ数値フォーマットで表示されます。分解能は 15 ビットです。使用される数値フォーマット例を下記の表に示します

入力電圧（－10V～0～10V）	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1111	7F FF	32767
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0003	0000 0000 0000 0001	00 01	1
0	0000 0000 0000 0000	0	0
-5	1100 0000 0000 0000	C000	49152
-10	1000 0000 0000 0000	8000	32768

バイナリーデータと入力電圧の関係

７５０－４７８：０～＋１０V
入力電圧値は16ビットのうちの15ビットを用いて表現します。

0 010　0000　0000　0000　（16ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 15ﾋﾞｯﾄを使います。

010　0000　0000　0000　（15ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

2　0　0　0　（16 進数）

↓

8192　（10 進数）

∴（8192 / 2^{15}）× 10V ＝ 2.5V

７５０－４７６：－１０V～＋１０V
＜データが＋側の場合＞

0 010　0000　0000　0000　（16ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 15ﾋﾞｯﾄを使います。

先頭は必ず「０」です。

010　0000　0000　0000　（15ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

2　0　0　0　（16 進数）

↓

8192　（10 進数）

∴（8192 / 2^{15}）× 10V ＝ 2.5V

＜データが－側の場合＞

先頭は必ず「１」です。マイナスを意味します。

1 100　0000　0000　0000　（16ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓データ 15 ビット分を反転します。

011　1111　1111　1111　(15 ビットバイナリ)

↓反転した１５ﾋﾞｯﾄを使います。

011　1111　1111　1111　（15ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

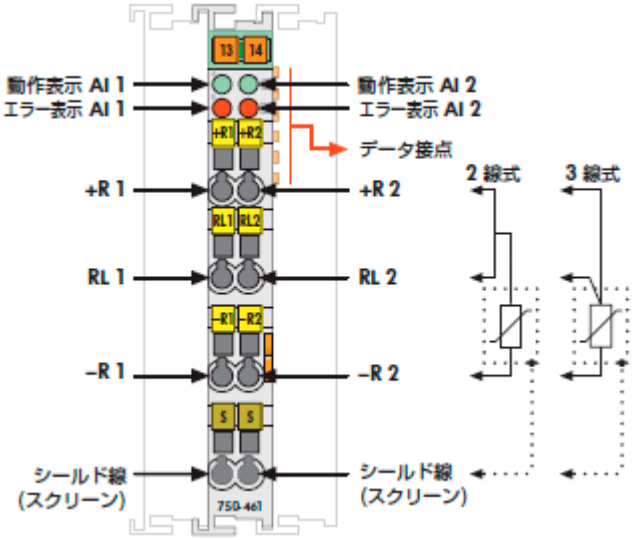
3　F　F　F　（16 進数）

↓

16383　（10 進数）

∴ －10 ×（16383 ＋ 1）/ 2^{15} ＝ －10 ×（16384 / 32768）＝ －5.0V

8.6 PT 100 用入力
型番 750-461 シリーズ



技術説明

750-461 シリーズは PT100、PT1000、Ni100、Ni1000 などの抵抗（RTD）センサを接続できるモジュールですが、標準仕様は PT100 タイプ用のものになります。PT100 以外のセンサタイプについては発注時にご指定ください。ご指定のない場合には PT100 タイプのもので納品されます。

各 RTD センサの種類に対して以下の型番のバリエーションがあります。

型番	表記	内容
750-461	2 点 PT100 入力モジュール	Pt100 用、－200℃～+850℃
750-461/000-003	2 点 PT1000 入力モジュール	Pt1000 用、－200℃～+850℃
750-461/000-006	2 点 PT100 入力モジュール、高精度	Pt100 用、－200℃～+850℃
750-461/000-004	2 点 Ni100 入力モジュール	Ni100 用、－60℃～+250℃
750-461/000-005	2 点 Ni1000 入力モジュール	Ni1000 用、－60℃～+250℃
750-461/003-000	2 点 PT100 入力モジュール、調整可	出荷時 Pt100 用、－200℃～+850℃

750-461 モジュールは、白金（Pt）またはニッケル（Ni）抵抗センサ（RTD）を直接結線することが可能です。2 線または 3 線の RTD 結線用に対応しています。

標準の 750-461 は、上記センサで規定されている-200°C から+850°C までの温度範囲で動作します。電圧分解能は 16 ビットで表わされます。A/D コンバータとプロセッサによって、電圧値が選択した測温抵抗センサの温度に比例したデジタル値に変換されます。

750-461/003-000 は WAGO-I/O-CHECK 2 ツールを用いて温度範囲などのパラメータ設定が可能なモジュールです。工場出荷時は Pt100 用になっています。

これらの I/O モジュールには電源ジャンパー接点が取付けられていません。駆動電源の供給は、データ接点から DC-DC コンバータを通して行われます。これらのモジュールでは内部からの電源によって動きます。

技術データ

型番 750-	461
入力数	2
入力電流（内部）	65mA
電源電圧	**システム電圧経由**
センサのタイプ	PT 100、PT 1000、Ni100、Ni1000（**発注時指定**）
接続電線数	**2 線、3 線（プリセット仕様）**
動作温度範囲	PT：-200°C～+850°C　　Ni：-60°C～250°C
分解能	**全領域で** 0.1°C
絶縁耐圧	500V（**システム－電源　間**）
測定電流	0.5mA type
内部ビット幅	2ch×16 **ビット**
測定精度	**25℃で**±0.2%**以下**
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；**0.08～2.5mm²
外形寸法（mm）　W×H×L	12×64*×100（***キャリアレールの上端から測定**）
プリセット仕様	**3 線の** PT 100
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)

マイクロプロセッサによって、全測定範囲におけるリニアライゼーション（直線化）が実行されます。ユーザは、上記にリストしている RTD タイプの温度範囲を使用することができます。これらのセンサの温度範囲は、1 ワード（16 ビット）で 0.1°C 当たり 1 ビットの分解能で表示されます。したがって、0°C が 16 進数値の 0000 に相当し、また 100°C が 03E8（デシマル 1000）に相当します。0°C よりも低い温度は、“1”で始まる 2 の補数フォーマットで表わされます。（くわしくは次頁をごらんください。）

短絡状態や RTD 線の切断が発生した場合には、赤色のエラー表示用 LED によって表示されます。緑色の LED が点灯しているときには、このモジュールと接続されたバスカップラー間の通信が正常に行われていることを示します。

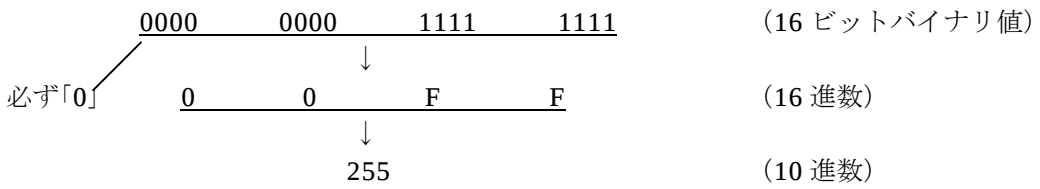
数値フォーマット

利用可能な数値範囲は、使用するセンサで標準化されている温度範囲を基準にすることになります。下記の表には、設定済みの PT 100 センサの数値フォーマットを記載しています。16 ビットバイナリデータを 10 進数に直したときにこの値が実温度（摂氏）の 10 倍になるように数値フォーマット化してあります。

温度（°C）	Pt100 抵抗（Ω）	Pt1000 抵抗（Ω）	バイナリ値	16 進数値	10 進数値
	＞400				
850	390.481	1384,998	0010 0001 0011 0100	2134	8500
100	138.506	1099,299	0000 0011 1110 1000	03E8	1000
25.5	109.929	1000,391	0000 0000 1111 1111	00FF	255
0.1	100.039	1000	0000 0000 0000 0001	0001	1
0	100	999,619	0000 0000 0000 0000	0000	0
-0.1	99.970	901,929	1111 1111 1111 1111	FFFF	-1
-25.5	90.389	184,936	1111 1111 0000 0001	FF01	-255
-200	18.192		1111 1000 0011 0000	F830	-2000
	＜18		1000 0000 0000 0000	8000	-32767

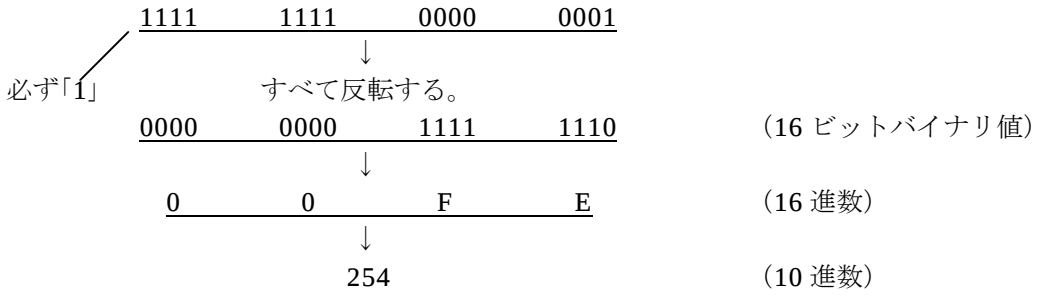
16 ビット数値フォーマットから温度を算出するには

① **温度がプラス側の場合**（16 ビット中の最高ビットは必ず「0」となります。）



∴ T ＝ 255 / 10 ＝ 25.5℃

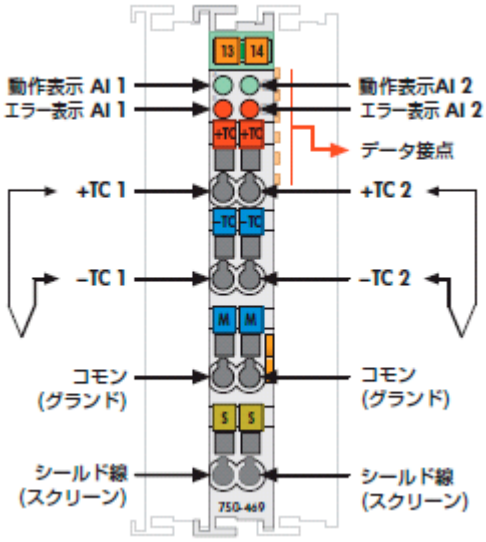
② **温度がマイナス側の場合**（16 ビット中の最高ビットは必ず「1」となります。）



この値に絶対値で 1 を加算して温度を算出する。

∴ T ＝-（254+1）/ 10 ＝-25.5℃

8.7　熱電対入力モジュール
型番 750-469（診断機能付）



技術説明

750-469 は各種熱電対に対応する豊富なバリエーションを取り揃えております。指定のない場合には標準仕様　K（-100～＋1370℃）タイプのもので納品されます。
一方、工場出荷後に仕様の変更ができる可変タイプとして 750-469/003-000 があります。実際に現場で用いられる熱電対の特性に合わせてモジュールの仕様を決めたい場合、ワゴジャパン内またはお客様のサイトでパラメータの設定をすることができます。このパラメータ設定には WAGO-I/O-CHECK 2 ソフトウェアツール（型番：759-302）を使用します。
WAGO-I/O-CHECK 2 の操作方法については以下の Web サイト内の 759-302 のマニュアルを参照ください。
http://www.wago.com/wagoweb/documentation/index_e.htm

シールドは DIN レールに結線されます。モジュールを DIN レール上にはめ込むと、各モジュール間のシールド結線が自動的に行われます。

注意
このモジュールには電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュール（例：2 チャンネルのデジタル入力モジュール）をこのモジュールの右側に接続する場合は、750-602 などの電源入力モジュールを挿入し、給電してください。

標準 K タイプの 750-469 および K タイプ以外の 750-469 枝番モジュールは　WAGO-I/O-SYSTEM のすべてのバスカプラ（低価格タイプを除く）／コントローラと共に使用することができます。

技術データ

型番　750-	469
入力数	2（差動入力、最大値：±3.5V）
電源電圧	システム電圧経由
センサのタイプ	K, S, T, J, E, L, mV 測定
冷接点補償	内部回路で補償
測定精度	25℃で±0.4%以下
分解能	1 ビットにつき 0.1°C
絶縁耐圧	500V（システム－電源）
入力電流（内部）	65mA 最大
内部ビット幅	2ch×16 ビット（断線検出付）
動作温度範囲	0°C～+55°C
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上部エッジから測定）
プリセット仕様	-100°C/+1370°C, Typ K
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-7
電磁波干渉に関する適合規格	EN50082-2(95) EN50081-2(94)

機能モジュール 750-469 では、熱電対センサを直接結線することが可能です。このモジュールは 2 線式熱電対の結線用に対応しています。2 線式の結線の場合には、熱電対線を+TC と-TC の各端子間に結線します。3 線式に対する結線の際には、シールドも結線します。

全温度範囲におけるリニアライゼーションと温度データへの変換はモジュール内のマイクロプロセッサによって実行されます。センサの検出温度範囲は、1 ワード（16 ビット）、0.1°C 当たり 1 ビットの分解能で表わされます。したがって、0°C が数値 0000 に相当し、また 25.5°C が数値 0x00FF に相当します。0°C よりも低い温度は、“1”で始まる 2 の補数フォーマットで表わされます。（「バイナリーデータと入力値の関係」を参考にしてください。）

すべての熱電対の規定動作温度範囲内で、機能モジュールは“µV メータ”と同様の働きを行います。電圧は 16 ビットの分解能で表わされます。プロセッサは、電圧値を、選択した熱電対タイプの測定温度に比例したデジタル値に変換します。

クランピングポイントにおけるオフセット電圧を補償するために、熱電対の冷接点補償計算が実行されます。内部回路にはその“ケージクランプ”結線部に温度測定用のセンサが内蔵されており、測定値の計算の際に温度オフセット電圧が自動的に補正されます。

750-469 の種類と結線可能なセンサの温度範囲

下記タイプを取り揃えておりますので、発注時に型名でご指定ください。

タイプ	温度範囲	指定型名
K	-100°C～+1370°C（標準タイプ）	750-469
S	0°C～+1700°C	750-469/000-001
T	-100°C～+400°C	750-469/000-002
J	-100°C～+1200°C	750-469/000-006
E	-100°C～1000°C	750-469/000-008
L	-100°C～+900°C	750-469/000-012
K（診断機能付）	-100°C～+1370°C	750-469/000-200
J（診断機能付）	-100°C～+1200°C	750-469/000-206
mV メータ	-120mV～+120mV	750-469/000-003

表 1：結線可能なセンサの温度範囲

LED 機能

緑色 LED：　機能

ON：　正常

OFF：　ウォッチドッグタイマーのオーバフロー

PLC から処理データが 100ms 間送信されなければ、緑色 LED の点灯が停止します。

赤色 LED：　エラー

ON：　測定温度範囲の上限または下限を超えた状態であるか、または断線を示します。

OFF：電圧が正常に測定範囲内に入っている。

断線検知機能

750-469 には診断機能を備えています。何らかの原因でセンサー（熱電対）が断線した場合あるいはセンサーが無結線の場合、出力データは自動的に「7FFF」（16 進数）になります。

数値フォーマット

温度データを示す数値はすべて一定の数値フォーマットで表わされます。デフォルト設定（K タイプの熱電対）では、1 ビットが 0.1°C に相当します。出力値は、規格に準拠して規定された各センサの温度範囲にそれぞれ対応します。コンフィギュレーションツールを使用して、出力フォーマットを選択することが可能です。リニアライゼーションをオフに切替えることが可能で、また基準温度の設定をオフに切替えることもできます。デフォルト設定範囲（K タイプ）での数値フォーマットを下記の表に記載します。

温度（°C）	電圧（uV）	バイナリ値	16 進数値	10 進数値
850	35314	0010 0001 0011 0100	2134	8500
100	4095	0000 0011 1110 1000	03E8	1000
25,5	1021	0000 0000 1111 1111	00FF	255
0,1	4	0000 0000 0000 0001	0001	1
0	0	0000 0000 0000 0000	0000	0
-0,1	-4	1111 1111 1111 1111	FFFF	-1
-25,5	-986	1111 1111 0000 0001	FF01	-255
-100	-3553	1111 1100 0001 1000	FC18	-1000

バイナリーデータから温度を算出するには

① **温度がプラス側の場合**（16 ビット中の最高ビットは必ず「0」となります。）

0000　0000　1111　1111　（16 ビットバイナリ値）

↓

必ず「0」　0　0　F　F　（16 進数）

↓

255　（10 進数）

∴ T ＝ 255 / 10 ＝ 25.5℃

② **温度がマイナス側の場合**（16 ビット中の最高ﾋﾞｯﾄは必ず「1」となります。）

1111　1111　0000　0001

↓

必ず「1」　すべて反転する。

0000　0000　1111　1110　（16 ビットバイナリ値）

↓

0　0　F　E　（16 進数）

↓

254　（10 進数）

この値に絶対値で 1 を加算して温度を算出する。

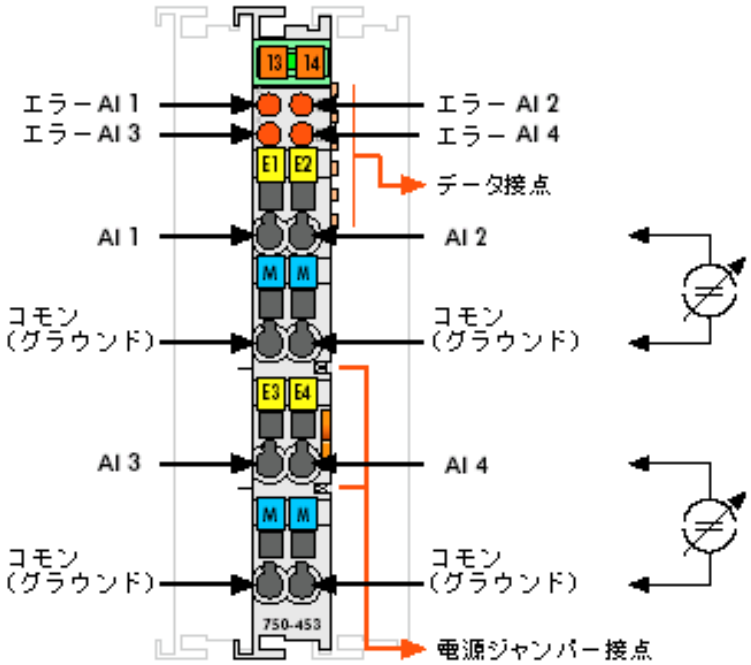
∴ T ＝-（254+1）/ 10 ＝- 25.5℃

750-469/000-003±120mV 入力の電圧入力モジュールビットフォーマット

電圧（mV）	バイナリ値	16 進数値	10 進数値
120.00	0100 1001 0011 1110	493E	18750
90.00	0011 0110 1110 1111	36EF	14063
60.00	0010 0100 1001 1111	249F	9375
30.00	0001 0010 0101 0000	1250	4688
0.00	0000 0000 0000 0000	0000	0
-30.00	1110 1101 1011 0001	EDB1	-4687
-60.00	1101 1011 0110 0001	DB61	-9375
-90.00	1100 1001 0001 0010	C912	-14062
-120.00	1011 0110 1100 0010	B6C2	-18750

wago_io_dll を使用した場合には 750-469 の温度値でデータが返ってきますので、このモジュールの場合はこれに「**0.0064**」を乗算してください。これでmV 値になります。

8.8　4 チャンネルアナログ入力 0～20mA/4～20mA （シングルエンド入力：外部電源供給型） 型番 750-453, 455



技術説明

入力チャンネルはシングルエンド入力で、コモングラウンド端子が用意されています。
入力は+I とコモンの各端子を通して結線します。外部で電源を準備する必要があります。すなわち電流を発生する電源回路が外部機器側で必要となります。

注意：
このモジュールの電源ジャンパー接点はモジュールの内部回路とは直接関係ありません。コモン電源をブリッジ（隣のモジュールへ橋渡し）する目的で設置されています。

技術データ

型番　750-	453	455
チャンネル数	4	
駆動電圧	システム電圧経由（DC-DC コンバータ）	
消費電流（内部）	60mA	
最大許容電圧	32V（最大）	
信号電流	0〜20mA	4〜20mA
入力インピーダンス	20mA 時に 100Ω以下	
分解能	12Bit	
絶縁耐圧	500V（システムー電源　間）	
変換時間	10ms（平均）	
内部ビット幅	4ch×16 ビット	
動作温度範囲	0°C〜+55°C	
測定精度	25°C 時フルスケールで±0.1%以内	
電線接続	ケージクランプ；0.08〜2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）	
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27	
電磁波干渉に関する適合規格	EN50082-2(96) EN50081-2(94)	

数値フォーマット

入力電流 0〜20mA	入力電流 4〜20mA	バイナリ値	16 進数値	10 進数値
20	20	0111 1111 1111 1000	7F F8	32760
10	12	0100 0000 0000 0000	40 00	16384
5	8	0010 0000 0000 0000	20 00	8192
2.5	6	0001 0000 0000 0000	10 00	4096
0.156	4.125	0000 0001 0000 0000	01 00	256
0.01	4.0078	0000 0000 0001 0000	00 10	16
0.005	4.0039	0000 0000 0000 1000	00 08	8
0	4	0000 0000 0000 0111	00 07	7
0	4	0000 0000 0000 0000	0	0

バイナリーデータから入力電流値を計算する。

① 750-453 の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0 （16 進数）

↓

1024 （10 進数）

$\therefore (1024 / 2^{12}) \times 20\text{mA} = 5\text{mA}$

② 750-455 の場合

0 010 0000 0000 0 000 （16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000 （12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

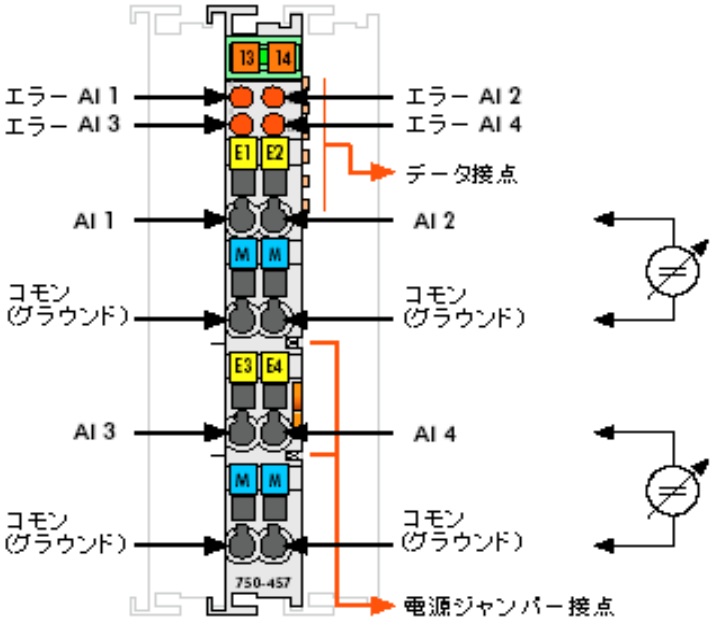
4 0 0 （16 進数）

↓

1024 （10 進数）

$\therefore 4\text{mA} + (1024 / 2^{12}) \times (20\text{mA} - 4\text{mA}) = 8\text{mA}$

8.9　4 チャンネルアナログ入力 ±10V/0-10V(シングルエンド入力)
型番 750-457,459



技術説明

入力チャンネルはシングルエンド入力で、コモングラウンド電位が用意されています。
4 チャンネルのコモングランドは共通で、お互いに誘導があります。

750-456,750-467 の廃価版で 4 チャンネルで構成されます。

尚、AD 変換の時間間隔は 10ms です。

注意：
このモジュールの電源ジャンパー接点はモジュールの内部回路とは直接関係ありません。コモン電源をブリッジ（隣のモジュールへ橋渡し）する目的で設置されています。

技術データ

型番　750-	457	459
チャンネル数	4	
駆動電圧	システム電圧経由（DC-DC コンバータ）	
消費電流（内部）	60mA	
最大定格電圧	±40V（最大）	
入力インピーダンス	100kΩ以下	
信号電圧	±10V	0V～10V
分解能	12Bit	
絶縁耐圧	500V（システム－電源間）	
変換時間	10ms	
ビット幅	4ch×16 ビット	
動作温度範囲	0°C～+55°C	
保存温度範囲	-25°C～+85°C	
測定精度	25°C 時フルスケールで±0.1%以内	
電線接続	ケージクランプ；0.08～2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）	
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27	
電磁波干渉に関する適合規格	EN50082-2(96) EN50081-2(94)	

数値フォーマット

７５０－４５７

入力電圧±10V	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32760
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0	0000 0000 0000 0000	0	00
-2.5	1110 0000 0000 0000	E0 00	57344
-5	1100 0000 0000 0000	C0 00	49152
-7.5	1010 0000 0000 0000	A0 00	40960
-10	1000 0000 0000 0000	80 00	32768

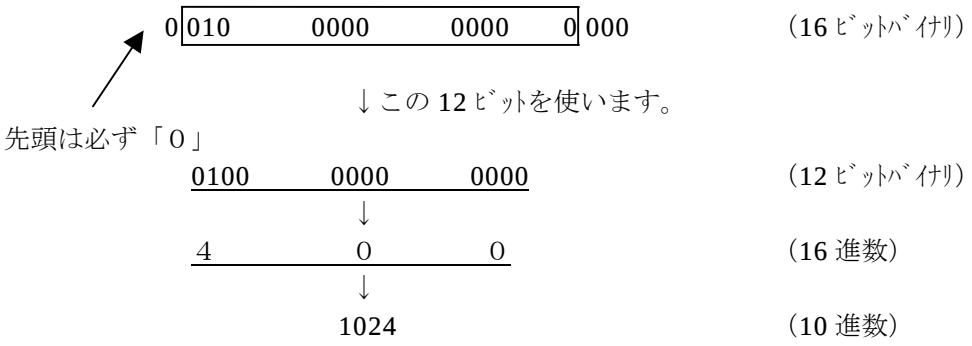
７５０－４５９

入力電圧（0～10V）	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32760
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 1000	00 08	8
0	0000 0000 0000 0111	00 07	7
0	0000 0000 0000 0000	0	0

バイナリーデータから入力電圧値を計算する。750-457

入力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

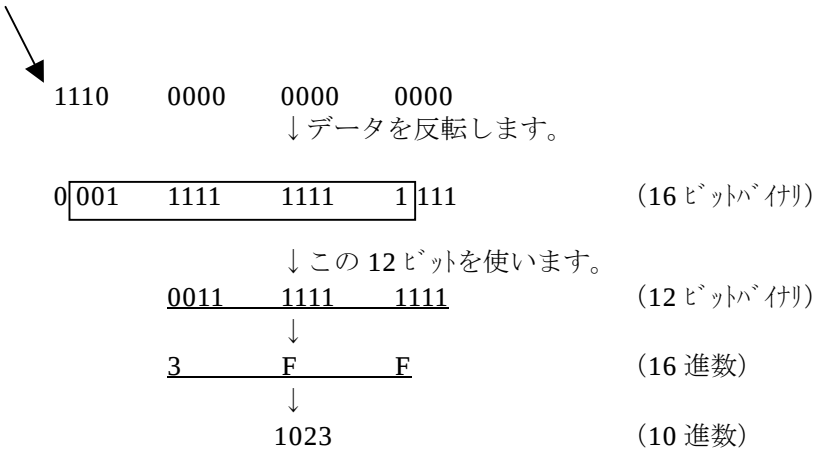
① データが＋側の場合



$\therefore (1024 / 2^{12}) \times 10\mathrm{V} = 2.5\mathrm{V}$

② データが－側の場合

先頭は必ず「1」です。



$\therefore -10 \times (1023 + 1) / 2^{12} = -1024 / 4096 = -2.5\mathrm{V}$

バイナリーデータから入力電圧値を計算する。750-459

入力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

0 010 0000 0000 0 000　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000　（12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0　（16 進数）

↓

1024　（10 進数）

∴ $(1024 / 2^{12}) \times 10V = 2.5V$

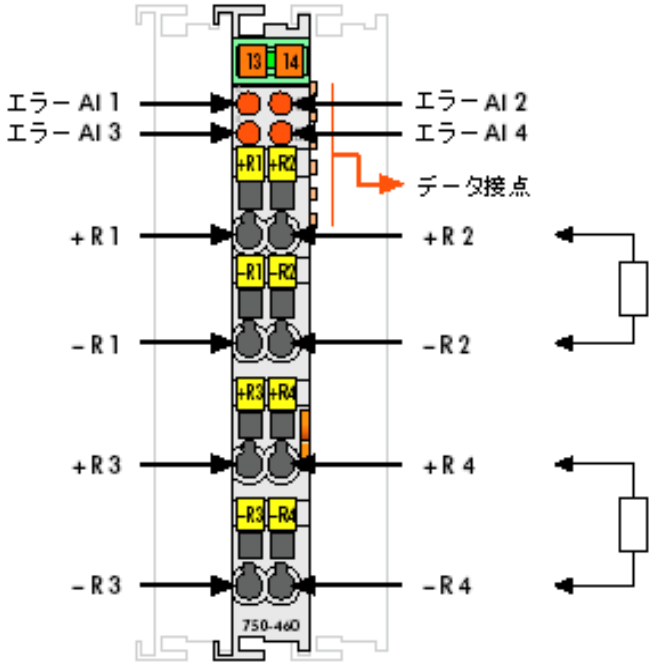
あるいは 12 ﾋﾞｯﾄバイナリデータから、

0100 0000 0000

2^{10}

∴ $(2^{10} / 2^{12}) \times 10V = 2.5V$

8.10　4 チャンネル RTD 用入力モジュール
型番 750-460　シリーズ



技術説明

750-460 モジュールは、白金抵抗のセンサを直接結線することが可能です。2 線の RTD 結線用に対応しています。

この機能モジュールは、PT 100 センサで規定されている-200°C から+850°C までの温度範囲で動作します。電圧分解能は 16 ビットで表わされます。A/D コンバータとプロセッサによって、電圧値が選択した測温抵抗センサの温度に比例したデジタル値に変換されます。

注意：
これらの I/O モジュールには電源ジャンパー接点が取付けられていません。駆動電源の供給は、データ接点から DC-DC コンバータを通して行われます。これらのモジュールでは内部からの電源によって動きます。

２線式の RTD センサーを接続する場合には、センサーからの距離について注意する必要があります。センサーとモジュール間の距離が長くなる場合にはこの抵抗分が負荷されますので、この調整をソフトウェア上で行ってください。

技術データ

型番 750-	460
入力数	4
入力電流（内部）	65mA
駆動電圧	**システム電圧経由（**DC-DC **コンバータ）**
センサのタイプ	PT 100、PT 200、 PT500、PT1000、Ni100**（発注時指定）**
接続電線数	2 **線式**
動作温度範囲	PT：-200°C～+850°C　　Ni：-60°C～+250°C
分解能	**全領域で** 0.1°C
絶縁耐圧	400V**（システム－電源　間）**
測定電流	0.5mA
ビット幅	4ch×16 **ビット**
測定精度	25**℃で**±1°C **以下**
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；**0.08～2.5mm^2
外形寸法（mm**）**　W×H×L	12×64*×100**（*キャリアレールの上端から測定）**
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN50082-2(96) EN50081-2(94)

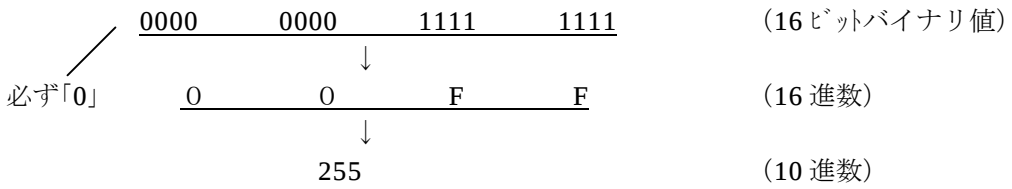
マイクロプロセッサによって、全測定範囲におけるリニアライゼーション（直線化）が実行されます。ユーザは、上記にリストしている RTD タイプの温度範囲を使用することができます。これらのセンサの温度範囲は、1 ワード（16 ビット）で 0.1°C 当たり 1 ビットの分解能で表示されます。したがって、0°C が 16 進数値の 0000 に相当し、また 100°C が 03E8（デシマル 1000）に相当します。0°C よりも低い温度は、“1”で始まる 2 の補数フォーマットで表わされます。（くわしくは次頁をごらんください。）

数値フォーマット

温度（°C）	Pt100 抵抗（Ω）	Pt1000 抵抗（Ω）	バイナリ値	16 進数値	10 進数値
	＞400				
850	390.481	1384,998	0010 0001 0011 0100	2134	8500
100	138.506	1099,299	0000 0011 1110 1000	03E8	1000
25.5	109.929	1000,391	0000 0000 1111 1111	00FF	255
0.1	100.039	1000	0000 0000 0000 0001	0001	1
0	100	999,619	0000 0000 0000 0000	0000	0
-0.1	99.970	901,929	1111 1111 1111 1111	FFFF	-1
-25.5	90.389	184,936	1111 1111 0000 0001	FF01	-255
-200	18.192		1111 1000 0011 0000	F830	-2000
	＜18		1000 0000 0000 0000	8000	-32767

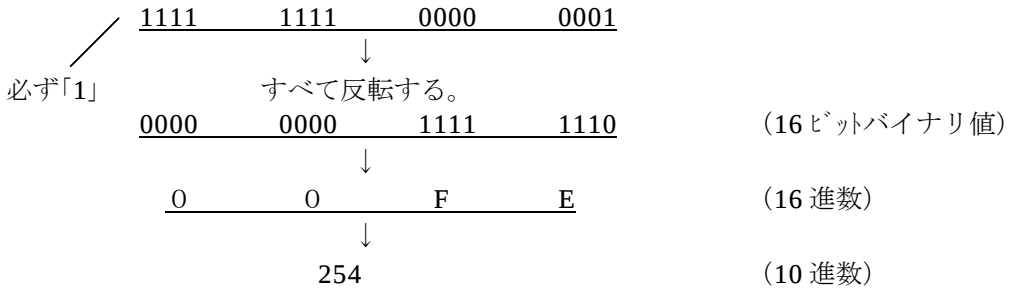
＜16 ビット数値フォーマットから温度を算出するには＞

① **温度がプラス側の場合**（16 ビット中の最高ビットは必ず「０」となります。）



∴ T ＝ 255 / 10 ＝ 25.5℃

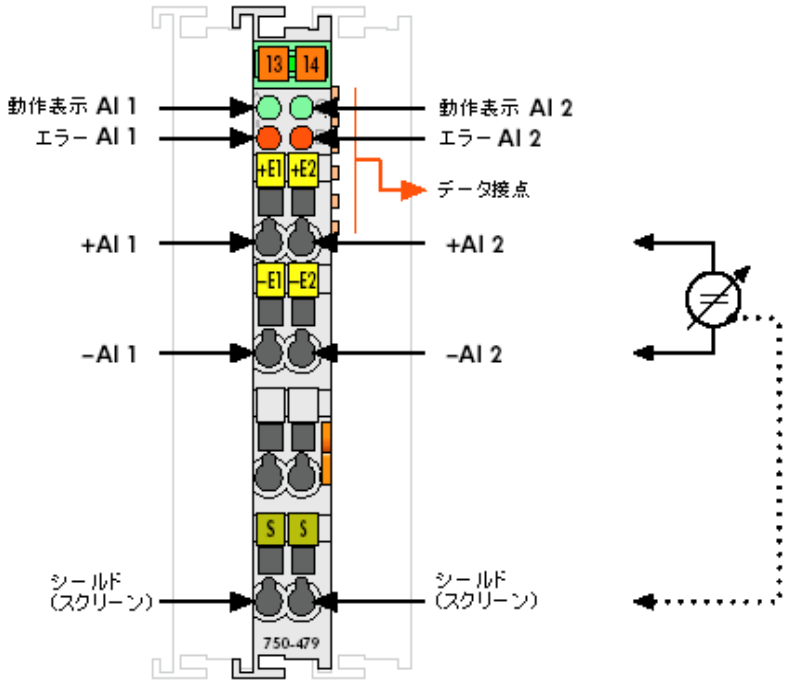
② **温度がマイナス側の場合**（16ﾋﾞｯﾄ中の最高ﾋﾞｯﾄは必ず「1」となります。）



この値に絶対値で１を加算して温度を算出する。

∴ T ＝ －（254 ＋ 1）/ 10 ＝ －25.5℃

8.11　2 チャンネルアナログ入力 ±10V（差動、外部電源利用型）
型番 750-479



技術説明

750-479 は±10V の 2 点電圧入力モジュールで、13Bit の分解能があります。

入力チャンネルは差動入力で、入力機器の電源については外部での駆動電源が必要となります。

入力は+I と－I の各端子を通して結線します。シールドは DIN レールに結線されます。すなわちモジュールを DIN レール上に嵌め込むと、各モジュール間のシールド接続が自動的に行われます。

技術データ

型番　750-	479
チャンネル数	2
駆動電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	100mA 以下
最大定格電圧	35V（最大）
入力信号電圧	±10V
入力インピーダンス	1 MΩ
分解能	13Bit およびサインビット
絶縁耐圧	500V（システムーチャンネル間、およびチャンネルーチャンネル間）
変換時間	2ms（サンプリング遅延時間含む）
クロストーク	-80dB 以下
チャンネル間遅延	1μ秒以下
ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
保存温度範囲	-25°C～+85°C
測定精度	25°C 時フルスケールで±0.05%以内
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN50082-2(96) EN50081-2(94)

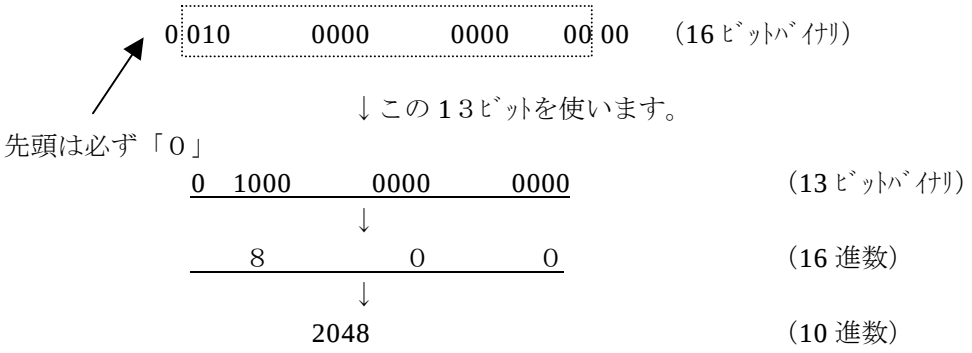
数値フォーマット

入力電圧±10V	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1100	7F FC	32764
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0012	0000 0000 0000 0100	00 04	4
0	0000 0000 0000 0011	0003	3
-2.5	1110 0000 0000 0000	E0 00	57344
-5	1100 0000 0000 0000	C0 00	49152
-7.5	1010 0000 0000 0000	A0 00	40960
-10	1000 0000 0000 0000	80 00	32768

バイナリーデータから入力電圧値を計算する。

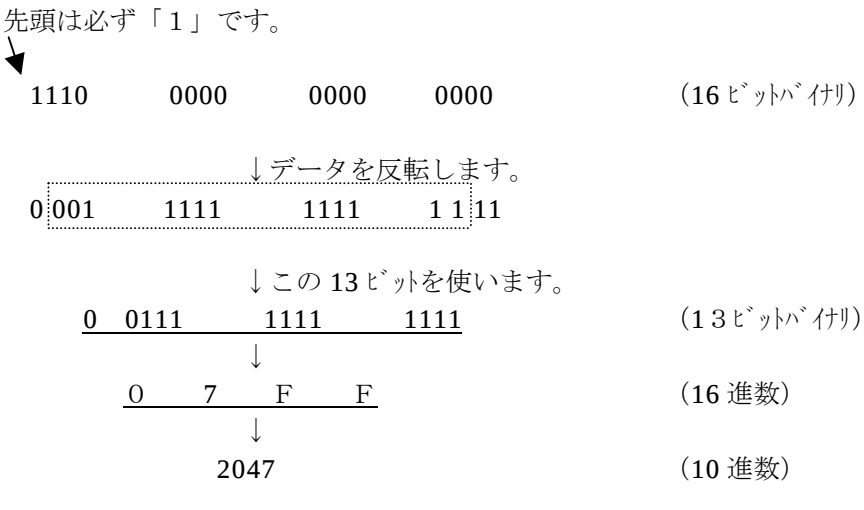
入力電圧値は 16 ビットのうちの１３ビットを用いて表現します。最下位の２ビットについては考慮しません。

① データが＋側の場合



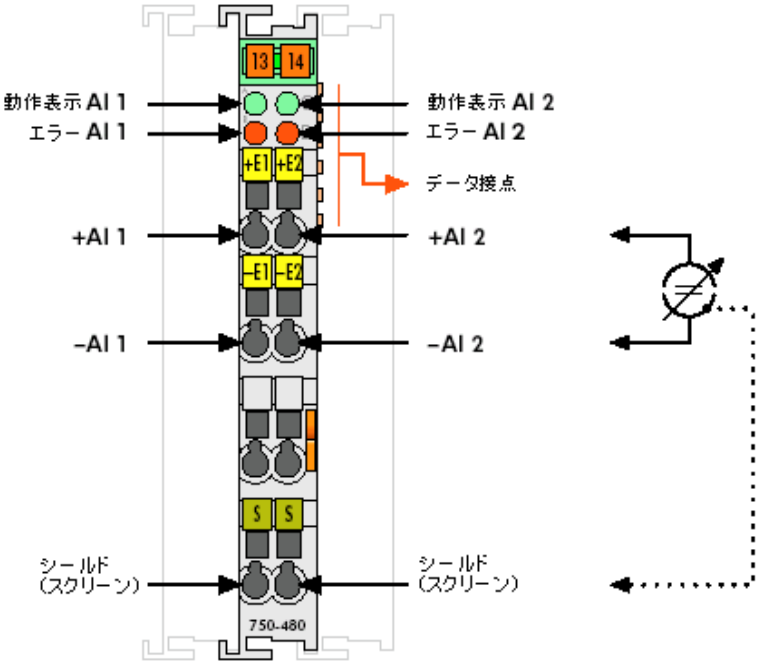
$\therefore (2024 / 2^{13}) \times 10\mathrm{V} = 2.5\mathrm{V}$

② データが－側の場合



$\therefore -10 \times (2047 + 1) / 2^{13} = -1024 / 4096 = -2.5\mathrm{V}$

8.12　2 チャンネルアナログ入力 0-20mA（差動、外部電源利用型）
型番 750-480



技術説明

750-480 は 0-20mA の 2 点電流入力モジュールで、13Bit の分解能があります。

入力チャンネルは差動入力で、チャンネル間は絶縁されています。両チャンネルのクロストークは-80dB 以下を実現しています。

入力インピーダンスは 20mA 時に 270Ω以下です。

入力は+I と－I の各端子を通して結線します。シールドは DIN レールに結線されます。すなわちモジュールを DIN レール上に嵌め込むと、各モジュール間のシールド接続が自動的に行われます。

このモジュールには電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュールは、このモジュールの右側に直接接続することはできません。３個の電源ジャンパー接点が用意されたモジュールを追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	480
チャンネル数	2
駆動電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	100mA 以上
許容連続負荷	30V
入力抵抗	20mA 時 270Ω以下
入力信号電流	0～20mA
分解能	13Bit
チャンネル間ｸﾛｽﾄｰｸ	－80ｄB 以下
変換時間	1ms
内部ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
保存温度範囲	-25°C～+85°C
測定精度	25°C フルスケールで±0.05%以内
電線接続	ケージクランプ；0.08～2.5mm²
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN50082-2(96) EN50081-2(94)

数値フォーマット

アナログ値はすべて数値単位フォーマットで表示されます。分解能は 13 ビットです。使用される数値フォーマットを下記の表に示します。3 個の最下位ビットについては考慮していません。

入力電流 0～20mA	バイナリ値	16 進数値	10 進数値
20	0111 1111 1111 1100	7F FC	32764
10	0100 0000 0000 0000	40 00	16384
5	0010 0000 0000 0000	20 00	8192
2.5	0001 0000 0000 0000	10 00	4096
0.156	0000 0001 0000 0000	01 00	256
0.0097	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 0100	00 04	4
0	0000 0000 0000 0011	00 03	3
0	0000 0000 0000 0000	0	0

バイナリデータと入力電流値の関係

入力電流値は 16 ビットのうちの 13 ビットを用いて表現します。最下位の 2 ビットについては考慮しません。

0 010 0000 0000 00 00　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 13 ﾋﾞｯﾄを使います。

0 1000 0000 0000　（13 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

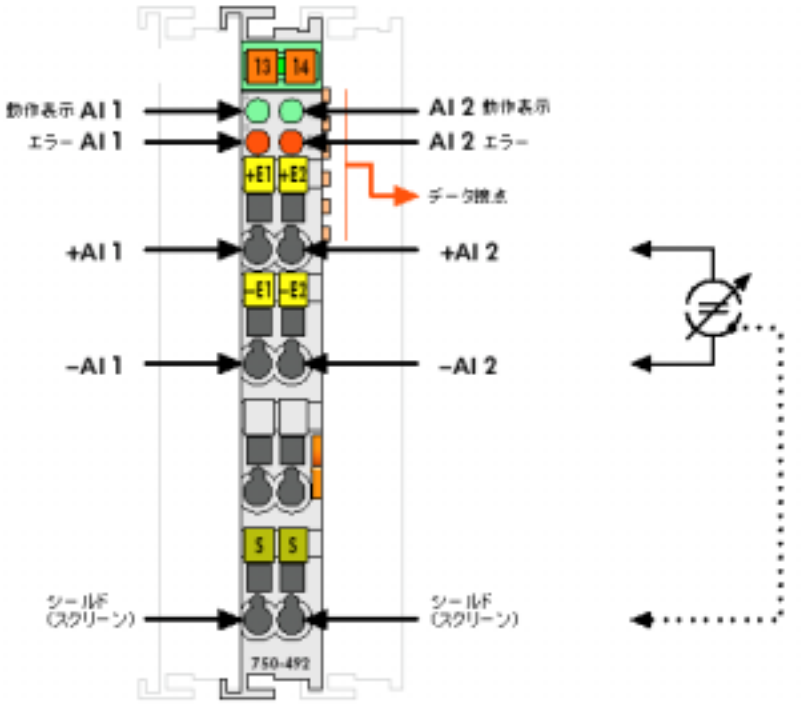
8 0 0　（16 進数）

↓

2048　（10 進数）

∴ $(2048 / 2^{13}) \times 20\text{mA} = 5\text{mA}$

8.13 2 チャンネルアナログ入力 4-20mA（絶縁差動、外部電源利用型）
型番 750-492



技術説明

入力チャンネルは差動入力で、チャンネル間は絶縁されています。両チャンネルのクロストークは-80dB 以下を実現しています。

入力インピーダンスは 20mA 時に 270Ω以下です。

入力は+I と－I の各端子を通して結線します。シールドは DIN レールに結線されます。すなわちモジュールを DIN レール上に嵌め込むと、各モジュール間のシールド接続が自動的に行われます。

このモジュールには電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュールは、このモジュールの右側に直接接続することはできません。３個の電源ジャンパー接点が用意されたモジュールを追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

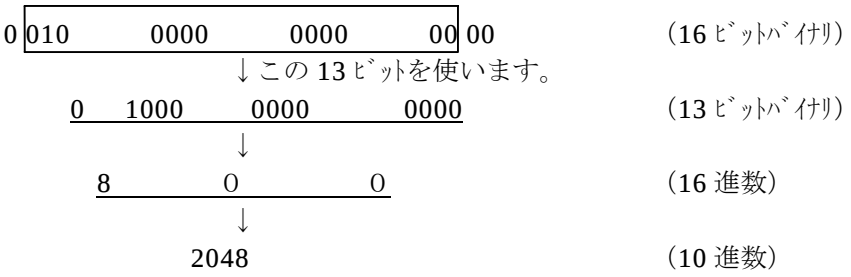
技術データ

型番　750-	492
チャンネル数	2
駆動電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	100mA 以上
許容連続負荷	30V
入力抵抗	20mA 時 270Ω以下
入力信号電流	4～20mA
分解能	13Bit
チャンネル間ｸﾛｽﾄｰｸ	－80ｄB 以下
変換時間	1ms
内部ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
保存温度範囲	-25°C～+85°C
測定精度	25°C フルスケールで±0.05%以内
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN50082-2(96) EN50081-2(94)

数値フォーマット

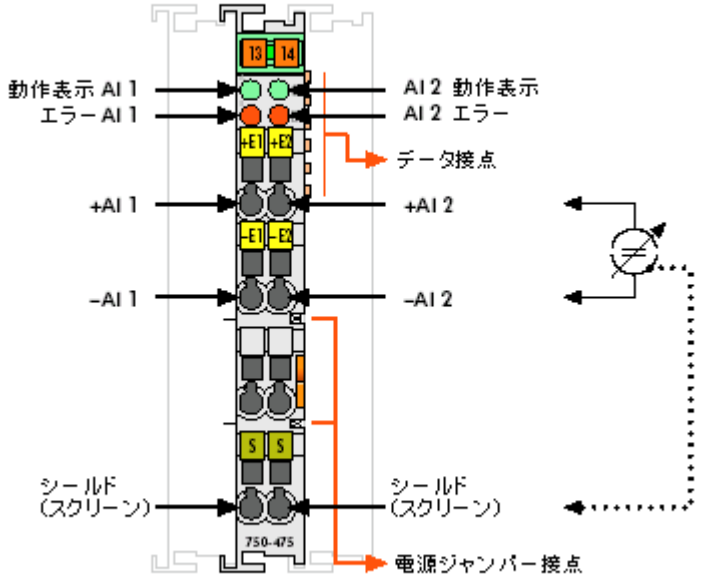
入力電流 (4～20mA)	バイナリ値	16 進数値	10 進数値
20	0111 1111 1111 1100	7F FC	32764
12	0100 0000 0000 0000	40 00	16384
8	0010 0000 0000 0000	20 00	8192
6	0001 0000 0000 0000	10 00	4096
4.125	0000 0001 0000 0000	01 00	256
4.0078	0000 0000 0001 0000	00 10	16
4.0019	0000 0000 0000 0100	00 04	8
4	0000 0000 0000 0011	00 03	3
4	0000 0000 0000 0000	0	0

バイナリーデータから入力電流値を計算する。



∴ **4mA ＋（2048 / 2^{13}）×（20mA－4mA）＝ 8mA**

8.14 2 チャンネルアナログ入力モジュール AC/DC 0-1A（差動、外部電源利用型） 型番 750-475



技術説明

モジュール 750-475 は 0 ～1A の電流信号入力モジュールです。入力点数として 1 チャンネルあたり 16 ﾋﾞｯﾄを占有しますがこのうち 12 ﾋﾞｯﾄを用いて入力電圧を表現します。

入力は+I と－I の各端子を通して結線します。シールドは DIN レールに結線されます。モジュールを DIN レール上に嵌め込むと、各モジュール間の電源結線が自動的に行われます。

電源の供給は、データ接点から DC-DC コンバータを通して行われます。この各モジュールは内部からの電源によって動作します。

注意：
このモジュールの電源ジャンパー接点はモジュールの内部回路とは直接関係ありません。コモン電源をブリッジ（隣のモジュールへ橋渡し）する目的で設置されています。

技術データ

型番　750-	475
チャンネル数	2
定格電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	80mA
最大許容電流	ピークで２A
信号電流	0A～1A 実効値(ピーク時は 2.0A)
入力抵抗	22mΩ
分解能	16Bit
絶縁耐圧	500V（システム－電源　間　実効値）
変換時間	200ms（平均）
内部ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
測定精度	25°C 時フルスケールで±0.1%以内
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN61000-6-2(00) EN50081-2(94)

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。分解能は 16 ビットです。使用される数値フォーマット例を下記の表に示します

入力電圧（0～2A）	バイナリ値	16 進数値	10 進数値
2	0100 1110 0010 0000	4E20	20000
1	0010 0111 0001 0000	27 10	10000
0.5	0001 0011 1000 1000	13 88	5000
0.25	0000 1001 1100 0100	09 C4	2500
0.1	0000 0000 0110 0400	00 64	100
100μA	0000 0000 0000 0001	00 01	1
0	0000 0000 0000 0000	0	0

バイナリーデータと入力電圧の関係

入力電流値は 16 ビットビットを用いて表現します。

0 001　　0011　　1000　　1000　　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

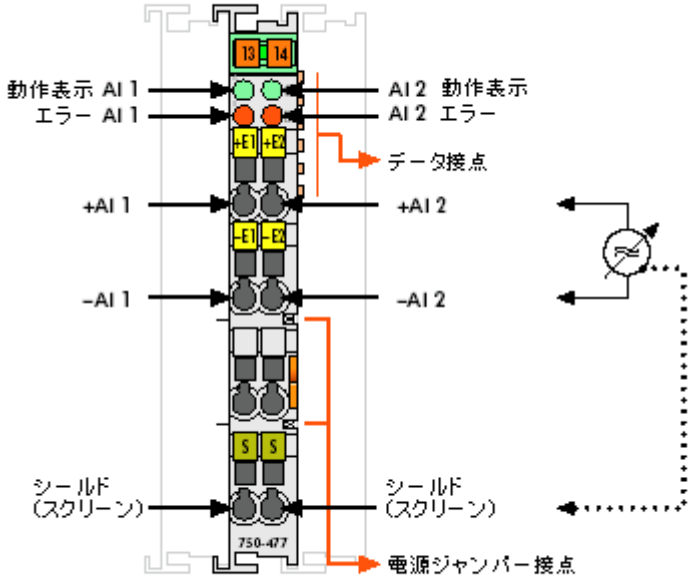
1　　3　　8　　8　　（16 進数）

↓

5000　　（10 進数）

∴ 5000 × 2 / 20000 A ＝ 0.5A

8.15 2 チャンネルアナログ入力モジュール AC/DC 0-10V（差動、外部電源利用型）型番 750-477



技術説明

モジュール 750-477 は AC/DC 対応、０～10V の電圧信号入力モジュールです。

入力チャンネルは差動入力です。

入力は＋I と－I の各端子を通して結線します。シールドは DIN レールに結線されます。モジュールを DIN レールに嵌め込むと、各モジュール間のシールド接続が自動的に行なわれます。

電源の供給は、データ接点から DC-DC コンバータを通して行われます。この各モジュールは内部からの電源によって動作します。

注意：
このモジュールの電源ジャンパー接点はモジュールの内部回路とは直接関係ありません。コモン電源をブリッジ（隣のモジュールへ橋渡し）する目的で設置されています。

技術データ

型番　750-	477
チャンネル数	2
定格電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	80mA
最大入力電圧	ピークで 20V
信号電圧	0V～10V 実効値（ピーク時は 20V）
入力インピーダンス	120kΩ
分解能	16Bit
絶縁耐圧	500V（システム－電源　間　実効値）
変換時間	200ms（平均）
内部ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
測定精度	25°C 時フルスケールで±0.1%以内
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN61000-6-2(00) EN50081-2(94)

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。分解能は 16 ビットです。使用される数値フォーマット例を下記の表に示します

入力電圧（0～20V）	バイナリ値	16 進数値	10 進数値
20	0100 1110 0010 0000	4E20	20000
10	0010 0111 0001 0000	27 10	10000
5	0001 0011 1000 1000	13 88	5000
2.5	0000 1001 1100 0100	09 C4	2500
1	0000 0000 0110 0400	00 64	100
1mV	0000 0000 0000 0001	00 01	1
0	0000 0000 0000 0000	0	0

バイナリーデータと入力電圧の関係

入力電圧値は 16 ビットを用いて表現します。

0 001　　0011　　1000　　1000　　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

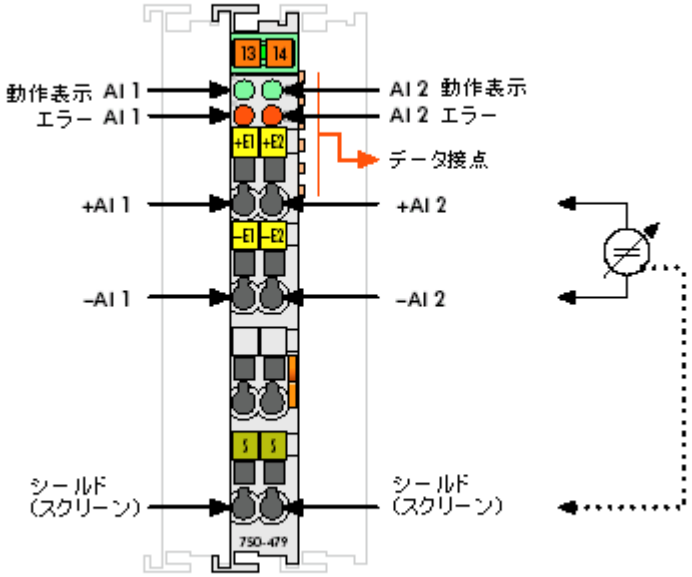
1　　3　　8　　8　　（16 進数）

↓

5000　　（10 進数）

∴ 5000 × 20 / 20000 A ＝ 5 V

8.16　2 チャンネルアナログ入力モジュール　0-30V（差動、外部電源利用型）
型番 750-483



技術説明

モジュール 750-483 は 0 ～30V の電圧信号入力モジュールです。入力点数として 1 チャンネルあたり 16 ﾋﾞｯﾄを占有しますがこのうち 14 ﾋﾞｯﾄを用いて入力電圧を表現します。

入力チャンネルは差動入力で、チャンネル間は絶縁されています。クロストークは-80dB 以下を実現しています。

この I/O モジュール用の電源の供給は、データ接点を通して行われます。

注意：
これらのモジュールには電源ジャンパー接点が用意されていません。電源ジャンパー接点が必要なモジュールをこのモジュールの右側に追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	483
チャンネル数	2、互いに電気的に絶縁
駆動電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	100mA 以内
最大許容連続電圧	230V
信号電圧	0V～30V
入力インピーダンス	1MΩ
分解能	14Bit
絶縁耐圧	500V（システムーチャンネル間、およびチャンネルーチャンネル間）
変換時間	2ms（サンプリング遅延時間含む）
クロストーク	-80dB 以下
チャンネル間遅延	1μ秒以下
ビット幅	2ch×16 ビット
動作温度範囲	0°C～+55°C
保存温度範囲	-25°C～+85°C
測定精度	25°C 時フルスケールで±0.05%以内
電線接続	ケージクランプ；0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレール上端から測定）
振動衝撃に関する適合規格	IEC60068-2-6 IEC60068-2-27
電磁波干渉に関する適合規格	EN61000-6-2(00) EN50081-2(94)

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。分解能は 14 ビットです。使用される数値フォーマット例を下記の表に示します 1 個の最下位ビットについては考慮していません。

入力電圧（0～30V）	バイナリ値	16 進数値	10 進数値
30	0111 1111 1111 1110	7F FE	32766
15	0100 0000 0000 0000	40 00	16384
7.4988	0010 0000 0000 0000	20 00	8192
3.7489	0001 0000 0000 0000	10 00	4096
0.2334	0000 0001 0000 0000	01 00	256
0.0137	0000 0000 0001 0000	00 10	16
0.0018	0000 0000 0000 0011	00 03	3
0	0000 0000 0000 0001	00 01	1
0	0000 0000 0000 0000	0	0

バイナリーデータと入力電圧の関係

入力電圧値は 16 ビットのうちの 14 ビットを用いて表現します。最下位の 1 ビットについては考慮しません。

0 010 0000 0000 000 0　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 14 ﾋﾞｯﾄを使います。

01 0000 0000 0000　（14 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

1 0 0 0　（16 進数）

↓

4096　（10 進数）

$\therefore (4096 / 2^{14}) \times 30V = 7.5V$

あるいは 14 ﾋﾞｯﾄバイナリデータから、

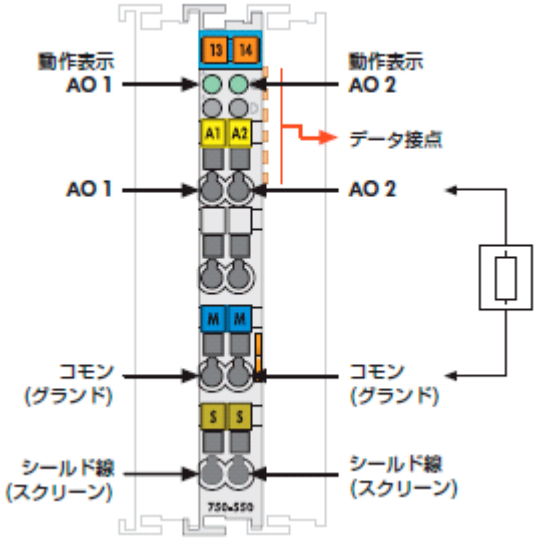
01 0000 0000 0000

2^{12}

$\therefore \quad (1 \times 2^{12} / 2^{14}) \times 30V = 7.5V$

9　アナログ出力モジュール

9.1　2 チャンネルアナログ出力 0～10V
型番 750-550



技術説明

モジュール 750-550 の出力信号は 0～10V の電圧信号です。1 チャンネルあたり 16 ﾋﾞｯﾄを占有しますが、このうち 12 ﾋﾞｯﾄを用いて出力電圧値を決定します。

この I/O モジュールには電源ジャンパー接点が用意されていません。電圧出力用駆動電源は、DC-DC コンバータを通して行われます。

注意：
このモジュール電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュールは、このモジュールの右側に直接接続することはできません。 3 個の電源ジャンパー接点が必要なモジュールを追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	550
チャンネル数	2
駆動電圧	システム電圧経由（DC-DC コンバータ）
消費電流（内部）	65mA
供給電圧	システム電圧経由
信号電圧	0〜10V
出力インピーダンス	5kΩ 以下
分解能	12Bit
絶縁耐圧	500V（システムー電源　間）
内部ビット幅	2ch×16 ビット
動作温度範囲	0°C〜+55°C
出力精度	25℃時±0.1%フルスケール以内 温度ドリフト：±0.01%/K 以内
電線接続	ケージクランプ；0.08〜2.5mm²
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）

数値フォーマット

アナログ値はすべてバイナリ単位フォーマットで表示されます。分解能は 12 ビットです。3 個の最下位ビットについては考慮していません。使用される数値フォーマットを下記の表に示します。

出力電圧 0〜10V	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32764
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 1000	00 08	8
0	0000 0000 0000 0111	00 07	7
0	0000 0000 0000 0000	0	0

バイナリーデータと出力電圧の関係

出力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

0 010 0000 0000 0 000　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100 0000 0000　（12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4 0 0　（16 進数）

↓

1 1024　（10 進数）

$\therefore\ (1024 / 2^{12}) \times 10V = 2.5V$

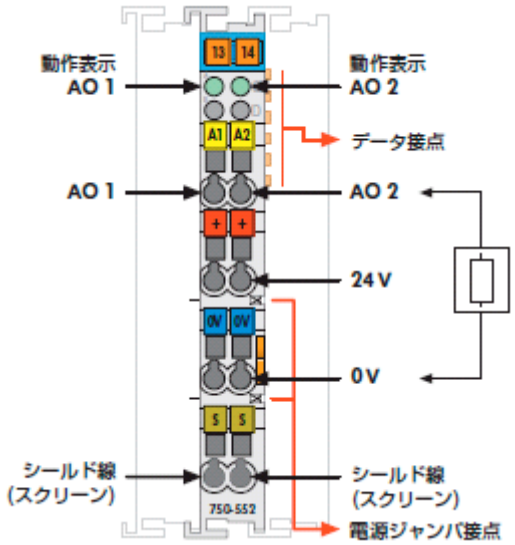
あるいは 12 ﾋﾞｯﾄバイナリデータから、

0100 0000 0000

2^{10}

$\therefore\ (2^{10} / 2^{12}) \times 10V = 2.5V$

9.2 2 チャンネルアナログ出力 0～20mA/4～20mA 型番 750-552, 554



技術説明

モジュール 750-552、554 の出力信号は 0～20mA または 4～20mA のプロセス信号です。シールドは DIN レールに結線されます。モジュールを DIN レール上に嵌め込むと、自動的にこの結線が行われます。

このモジュールはセンサー用電源ジャンパーから供給される 24 V を用いて外部回路等を駆動します。電源ジャンパーが左側から続いていない場合は新たに電源モジュール（例えば、モジュール 750-602 など）を接続して電源の供給を行う必要があります。

技術データ

型番　750-	552	554
チャンネル数	2	
消費電流（内部）	60mA（最大）	
駆動電圧	24V DC（-15%/+20%）電源ジャンパー接点経由	
信号電流	0〜20mA	4〜20mA
出力インピーダンス	500Ω以下	
分解能	12Bit	
絶縁耐圧	500V（システム／電源　間）	
内部ビット幅	2ch×16 ビット	
動作温度範囲	0°C〜+55°C	
出力精度	フルスケール±0.5%以内	
電線接続	ケージクランプ；0.08〜2.5mm^2	
外形寸法（mm）　W×H×L	12×64*×100 （*キャリアレールの上端から測定）	

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。使用される数値フォーマットを下記の表に示します 3 個の最下位ビットについては考慮していません。

出力電流 （0〜20mA）	出力電流 （4〜20mA）	バイナリ値	16 進数値	10 進数値
20	20	0111 1111 1111 1000	7F F8	32767
10	12	0100 0000 0000 0000	40 00	16384
5	8	0010 0000 0000 0000	20 00	8192
2.5	6	0001 0000 0000 0000	10 00	4096
0.156	4.125	0000 0001 0000 0000	01 00	256
0.01	4.0078	0000 0000 0001 0000	00 10	16
0.005	4.0039	0000 0000 0000 1000	00 08	8
0	4	0000 0000 0000 0111	00 07	7
0	4	0000 0000 0000 0000	0	0

バイナリデータと出力電流の関係

出力電流値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

① 750-552 の場合

0 010　　0000　　0000　　0 000　　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100　　0000　　0000　　（12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4　　0　　0　　（16 進数）

↓

1024　　（10 進数）

$\therefore (1024 / 2^{12}) \times 20\text{mA} = 5\text{mA}$

② 750-554 の場合

0 010　　0000　　0000　　0 000　　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100　　0000　　0000　　（12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

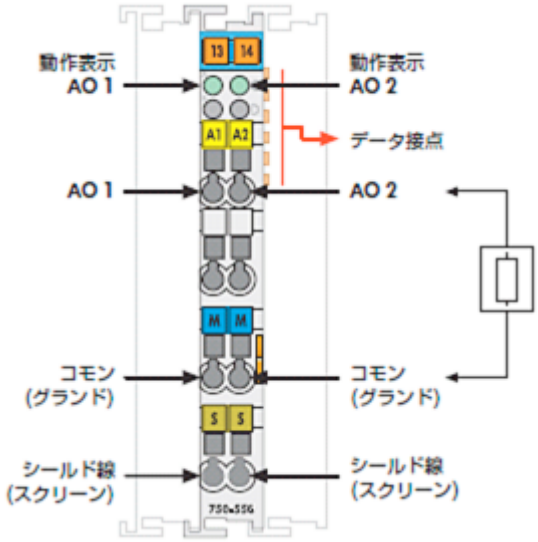
4　　0　　0　　（16 進数）

↓

1024　　（10 進数）

$\therefore 4\text{mA} + (1024 / 2^{12}) \times (20\text{mA} - 4\text{mA}) = 8\text{mA}$

9.3　2 チャンネルアナログ出力+/-10V
型番 750-556



技術説明

モジュール 750-556 の出力信号は±10V 電圧信号です。コモングラウンドに対する電位で出力されます。

シールドは DIN レールに直結します。モジュールを DIN レール上に嵌め込むと、自動的にこの結線が行われます。
これらの I/O モジュールには電源ジャンパー接点が用意されておらず、駆動電源の供給は、DC-DC コンバータを通して行われます。

注意：
このモジュールには電源ジャンパー接点が用意されていません。したがって、電源ジャンパー接点が必要なモジュールをこの右側に直接接続することはできません。この場合は電源モジュール（750-602）などを接続してください。

技術データ

型番　750-	556
チャンネル数	2
定格電圧	**システム電圧経由（DC-DC コンバータ）**
消費電流（内部）	65mA
信号電圧	±10 V
出力インピーダンス	5kΩ**以下**
分解能	12Bit
絶縁耐圧	500V**（システムー電源　間）**
内部ビット幅	2ch×16 **ビット**
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；**0.08～2.5mm^2
外形寸法（mm**）　**W×H×L	12×64*×100**（*キャリアレールの上端から測定）**

数値フォーマット

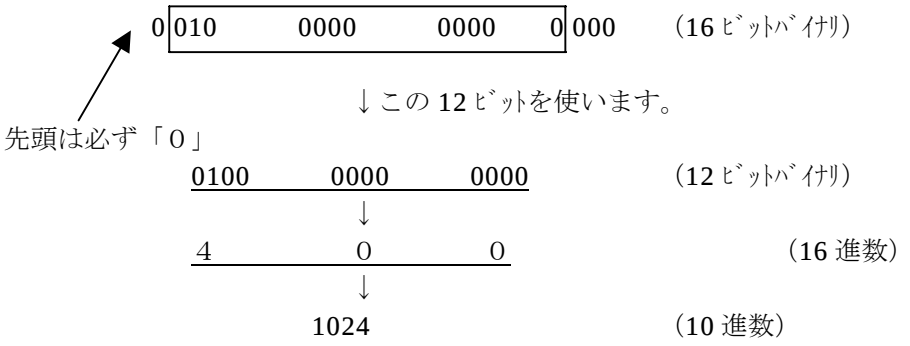
アナログ値はすべて数値単位フォーマットで表示されます。分解能は 12 ビットです。3 個の最下位ビットは無視されます。使用される数値フォーマットを下記の表に示します。

出力電圧（±10V**）**	**バイナリ値**	16 **進数値**	10 **進数値**
10	0111 1111 1111 1000	7F F8	32767
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 1000	00 08	8
0	0000 0000 0000 0111	7	7
-2.5	1110 0000 0000 0000	E0 00	57344
-5	1100 0000 0000 0000	C0 00	49152
-7.5	1010 0000 0000 0000	A0 00	40960
-10	1000 0000 0000 0000	80 00	32768

バイナリーデータと出力電圧値の関係

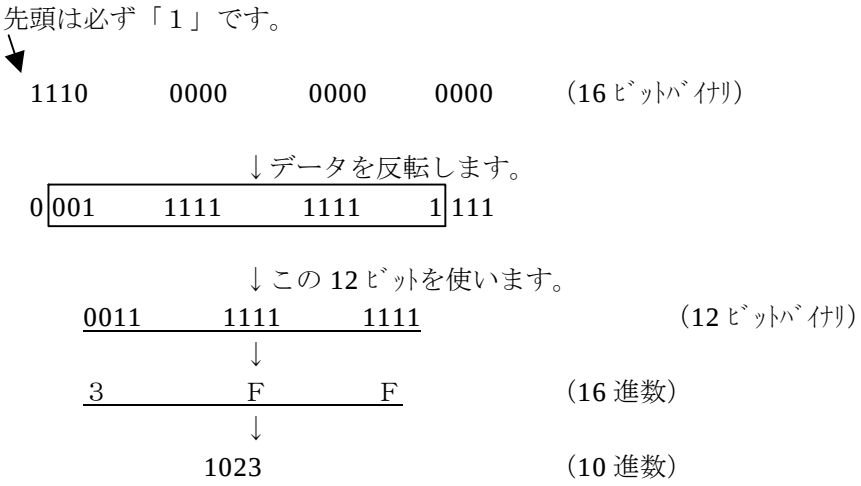
出力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

① 正電圧の出力



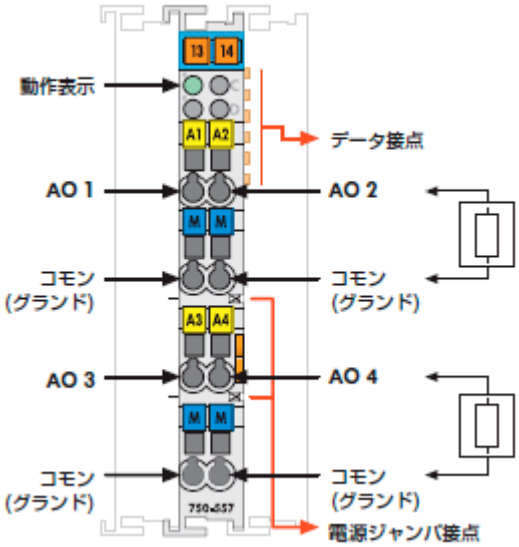
∴ $(1024 / 2^{12}) \times 10\text{V} = 2.5\text{V}$

② 負電圧の出力



∴ $-10\times(1023 + 1) / 2^{12} = -1024 / 4096 = -2.5\text{V}$

9.4　4 チャンネルアナログ出力 ±10V/0-10V
型番 750-557,559



技術説明

モジュール 750-557 の出力信号は±10V の電圧信号、750-559 は 0～10V の電圧信号です。１チャンネルあたり１６ﾋﾞｯﾄを占有しますが、このうち１２ﾋﾞｯﾄを用いて出力電圧値を決定します。

電圧出力用駆動電源は、DC-DC コンバータを通してデータ接点から供給されます。

また、このモジュールは４ch 分のデータをマルチプレクスして出力しているので、各チャンネルスキャンごとに出力フィルタを周期設定しています。１周期に約 100ｍS を必要とします。

注意：
３個の電源ジャンパー接点が必要なモジュールを追加するためには 750-601 などの電源モジュールを挿入して電源を外部から再供給してください。

技術データ

型番　750-	557	559
チャンネル数（入力数）	4	
駆動電圧	システム電圧経由（DC-DC コンバータ）	
消費電流（内部）	125mA	
供給電圧	システム電圧経由	
信号電圧	±10V	0～10V
出力インピーダンス	5kΩ 以下	
分解能	12Bit	
絶縁耐圧	500V（システムー電源　間）	
内部ビット幅	4ch×16 ビット	
動作温度範囲	0°C～+55°C	
変換時間	10ms	
フィルター回路リセット時間.	100ms	
25°C時測定誤差	フルスケール±0.1% 以内	
測定誤差の温度ドリフト	フルスケール±0.01%/K 以内	
電線接続	ケージクランプ；0.08～2.5mm²	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）	

数値フォーマット　750-557 (±10V)

７５０－５５７

出力電圧（±10V）	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32767
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 1000	00 08	8
0	0000 0000 0000 0111	7	7
-2.5	1110 0000 0000 0000	E0 00	57344
-5	1100 0000 0000 0000	C0 00	49152
-7.5	1010 0000 0000 0000	A0 00	40960
-10	1000 0000 0000 0000	80 00	32768

バイナリーデータと出力電圧値の関係　750-559 (±10V)

出力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

① 正電圧の出力

0 010　　0000　　0000　　0 000　　(16 ﾋﾞｯﾄﾊﾞｲﾅﾘ)

↓この 12 ﾋﾞｯﾄを使います。

先頭は必ず「0」

0100　　0000　　0000　　(12 ﾋﾞｯﾄﾊﾞｲﾅﾘ)

↓

4　　0　　0　　(16 進数)

↓

1024　　(10 進数)

$\therefore (1024 / 2^{12}) \times 10V = 2.5V$

② 負電圧の出力

先頭は必ず「1」です。

1110　　0000　　0000　　0000　　(16 ﾋﾞｯﾄﾊﾞｲﾅﾘ)

↓データを反転します。

0 001　　1111　　1111　　1 111

↓この 12 ﾋﾞｯﾄを使います。

0011　　1111　　1111　　(12 ﾋﾞｯﾄﾊﾞｲﾅﾘ)

↓

3　　F　　F　　(16 進数)

↓

1023　　(10 進数)

$\therefore -10 \times (1023 + 1) / 2^{12} = -1024 / 4096 = -2.5V$

数値フォーマット　750-559 (0～10V)

アナログ値はすべてバイナリ単位フォーマットで表示されます。分解能は 12 ビットです。3 個の最下位ビットについては考慮していません。使用される数値フォーマットを下記の表に示します。

７５０－５５９

出力電圧 0～10V	バイナリ値	16 進数値	10 進数値
10	0111 1111 1111 1000	7F F8	32767
5	0100 0000 0000 0000	40 00	16384
2.5	0010 0000 0000 0000	20 00	8192
1.25	0001 0000 0000 0000	10 00	4096
0.0781	0000 0001 0000 0000	01 00	256
0.0049	0000 0000 0001 0000	00 10	16
0.0024	0000 0000 0000 1000	00 08	8
0	0000 0000 0000 0111	00 07	7
0	0000 0000 0000 0000	0	0

バイナリーデータと出力電圧の関係 (750-559)

出力電圧値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

0 010　　0000　　0000　　0 000　　（16 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓この 12 ﾋﾞｯﾄを使います。

0100　　0000　　0000　　（12 ﾋﾞｯﾄﾊﾞｲﾅﾘ）

↓

4　　0　　0　　（16 進数）

↓

1024　　（10 進数）

$\therefore\ (1024 / 2^{12}) \times 10V = 2.5V$

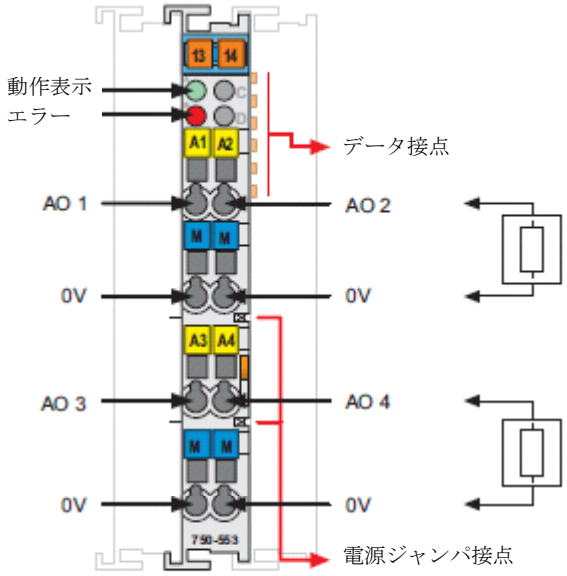
あるいは 12 ﾋﾞｯﾄバイナリデータから、

0100　0000　0000

2^{10}

$\therefore\ (2^{10} / 2^{12}) \times 10V = 2.5V$

9.5 4 チャンネルアナログ出力 0～20mA/4～20mA 型番 750-553, 555



技術説明

750-553 および 750-555 は出力信号が 0～20mA または 4～20mA のプロセス信号を出力するモジュールです。このモジュールは 4 組の出力端子とコモン端子を持っており、2 線式アクチュエータを直接 AO1～AO4（信号出力）と各コモン（グランド）に接続できます。

注意

一つの I/O モジュールの出力に接続された全ての負荷インピーダンスは 0～300 Ω または 300～600 Ω の範囲になければなりません。負荷インピーダンスはこのモジュールでチェックされ、許容外の負荷の組み合わせはエラーを表示することになります。

入力信号は電気的に絶縁され、12 ビットの分解能で転送されます。
動作が正常であり、内部バス通信のエラーがなければ緑色の LED が点灯します。一方、断線や負荷の組み合わせ異常があった場合は赤色のエラーLED が点灯します。

外部機器への電源供給は、モジュールを DIN レール上に嵌め込むと電源ジャンパー接点を通して行われます。電源ジャンパー接点のないモジュールが左側にあった場合は、電源入力モジュール（例えば、モジュール 750-602 など）を接続して電源の供給を行う必要があります。

これらの出力モジュールは WAGO-I/O-SYSTEM のすべてのバスカプラ（低価格タイプを除く）／コントローラに接続することが可能です。

技術データ

型番　750-	553	555
チャンネル数	4	
消費電流（内部）	60mA（最大）	
定格電圧	24V DC（-15%/+20%）電源ジャンパー接点経由	
信号電流	0～20mA	4～20mA
負荷インピーダンス	0～300Ωまたは 300～600Ω （全ての負荷は同じ範囲の負荷であること）	
変換時間	平均 10ms	
出力フィルタ設定時間	平均 100ms	
測定誤差（25℃）	フルスケール値の±0.1%以下	
温度係数	フルスケール値の±0.01%/K 以下	
分解能	12Bit	
絶縁耐圧	500V システム／電源	
ビット幅	4ch×16 ビットデータ 4ch×16 ビット制御／ステータス（オプション）	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm^2	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアレールの上端から測定）	

数値フォーマット

アナログ値はすべてバイナリ数値フォーマットで表示されます。使用される数値フォーマットを下記の表に示します（モジュール 750-553/555）。3 個の最下位ビットについては考慮していません。

出力電流（0～20mA）	出力電流（4～20mA）	バイナリ値	16 進数値	10 進数値	ステータスバイト*
20	20	0111 1111 1111 1111	7F FF	32767	0x00
10	12	0100 0000 0000 0000	40 00	16384	0x00
5	8	0010 0000 0000 0000	20 00	8192	0x00
2.5	6	0001 0000 0000 0000	10 00	4096	0x00
0.156	4.125	0000 0001 0000 0000	01 00	256	0x00
0.01	4.0078	0000 0000 0001 0000	00 10	16	0x00
0.005	4.0039	0000 0000 0000 1000	00 08	8	0x00
0	4	0000 0000 0000 0111	00 07	7	0x00
0	4	0000 0000 0000 0000	0	0	0x00
断線または許容外の負荷組み合わせ		XXXX.XXXX.XXXX.XXXX	XXXX	X	0x40

*ステータスバイトの有無はフィールドバスの種類によって異なり、オプションです。

バイナリデータと出力電流の関係

出力電流値は 16 ビットのうちの 12 ビットを用いて表現します。最下位の 3 ビットについては考慮しません。

① 750-553 の場合

0 010　0000　0000　0 000　（16 ビットバイナリ）

↓この 12 ビットを使います。

0100　0000　0000　（12 ビットバイナリ）

↓

4　0　0　（16 進数）

↓

1024　（10 進数）

$\therefore \quad (1024 / 2^{12}) \times 20\text{mA} = 5\text{mA}$

② 750-555 の場合

0 010　0000　0000　0 000　（16 ビットバイナリ）

↓この 12 ビットを使います。

0100　0000　0000　（12 ビットバイナリ）

↓

4　0　0　（16 進数）

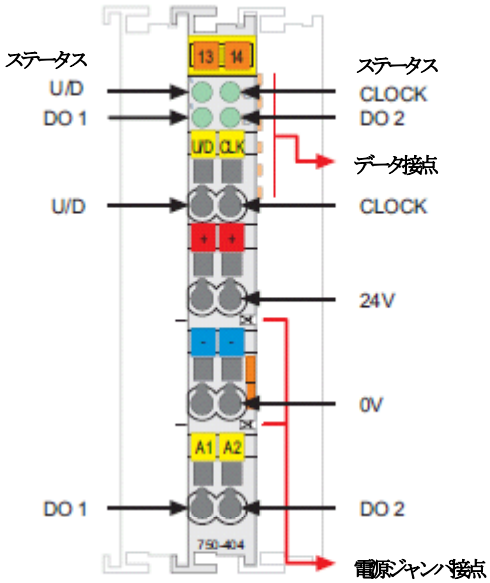
↓

1024　（10 進数）

$\therefore \ 4\text{mA} + (1024/2^{12}) \times (20\text{mA} - 4\text{mA}) = 8\text{mA}$

10 カウンタモジュール

10.1 型番 750-404, 750-404/000-001, 750-404/000-002
アップ／ダウンカウンタ 100kHz, 750-404



概要

このカウンタモジュールはクロックの入力でカウントを行います。32 ビットのカウンタを内蔵しています。

このセクションでは、アップ／ダウン入力を備えたカウンタの機能および動作について説明します。

カウンタモジュールはすべての WAGO-I/O-SYSTEM のバスカプラ（低価格タイプを除く）とともに使用して動作させることが可能です。

技術データ

型番　750-	404, 404/000-001	404/000-002
出力数	2	
出力電流	0.5A	
カウンタ数	1	
消費電流（内部）	70mA（平均）	
定格電圧	24V DC（-15%/+20%）	
信号電圧（0）	-3V～+5V DC	
信号電圧（1）	+15V～+30V DC	
スイッチング速度	100kHz	1kHz（最大）
出力電流	5mA	
カウンタサイズ	32Bit	
絶縁耐圧	500V システム／電源	
ビット幅	32 ビット（8 ビット制御用；8 ビットステータス用）	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm^2	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	

プロセスイメージ

このモジュールは出力および入力のためにそれぞれ 6 バイト（48 ビット、3 ワード分）のバッファを持っています。このうち 1 バイト分は制御（ステータス）バイトで、カウンタ値の設定および読出しは 32 ビット（4 バイト分）となります。
バスカプラ／コントローラ内の 750-404 のプロセスイメージは以下のとおりです。

出力データ	上位バイト	下位バイト
出力ワード 1	空き	制御バイト
出力ワード 2	カウンタ②　9～16 ビット	カウンタ①　1～8 ビット
出力ワード 3	カウンタ④　25～32 ビット	カウンタ③　17～24 ビット

入力データ	上位バイト	下位バイト
入力ワード 1	空き	ステータスバイト
入力ワード 2	カウンタ②　9～16 ビット	カウンタ①　1～8 ビット
入力ワード 3	カウンタ④　25～32 ビット	カウンタ③　17～24 ビット

機能説明

このカウンタはクロック入力の信号パルスでカウント処理動作を開始します。このパルスが 0V から 24V へ変化したときカウントされます。

入力 U/D が 24V に設定されると、このカウンタはカウントアップ動作を行います。入力がオープン回路になった場合または 0V のときは、逆方向のカウント動作（カウントダウン）が行われます。

モジュールの最下部にある 2 個の接続部には外部の出力がそれぞれ用意されています。これらの出力は、制御バイトの Bit2、Bit3 の設定によってオン／オフすることができます。

フィールドバス上のマスタ（PLC など）から本カウンタモジュールにカウンタ値を設定する場合やモジュール内の現在カウンタ値を読み出す場合は次に示すようなデータフォーマットになります。
以下の出力データ（カウンタ値の設定）および入力データ（カウンタ値の読出し）のフォーマットは Ethernet、DeviceNet、CC-Link の各バスカプラ／コントローラに共通です。

出力データ（カウンタ値の設定）

出力ワード 1		出力ワード 2		出力ワード 3	
使用せず（空き）	制御バイト	カウンタ② 9〜16 ビット	カウンタ① 1〜8 ビット	カウンタ④ 25〜32 ビット	カウンタ③ 17〜24 ビット

入力データ（カウンタ値の読出し）

入力ワード 1		入力ワード 2		入力ワード 3	
使用せず（空き）	ステータスバイト	カウンタ② 9〜16 ビット	カウンタ① 1〜8 ビット	カウンタ④ 25〜32 ビット	カウンタ③ 17〜24 ビット

Profibus に関しては以下のフォーマットになります。

出力データ（カウンタ値の設定）

出力ワード 1		出力ワード 2		出力ワード 3	
使用せず（空き）	制御バイト	カウンタ③ 17〜24 ビット	カウンタ④ 25〜32 ビット	カウンタ① 1〜8 ビット	カウンタ② 9〜16 ビット

入力データ（カウンタ値の読出し）

入力ワード 1		入力ワード 2		入力ワード 3	
使用せず（空き）	ステータスバイト	カウンタ③ 17〜24 ビット	カウンタ④ 25〜32 ビット	カウンタ① 1〜8 ビット	カウンタ② 9〜16 ビット

出力データ中の制御バイトは下記の各ビットによって構成されています。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ設定 リセット	カウンタ停止	出力 O2 をアクティブ(オン)にする	出力 O1 をアクティブ(オン)にする	0 不使用	0 不使用

入力データ中のステータスバイトは下記の各ビットによって構成されます。

ステータスバイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ設定中	カウンタが停止	出力 O2 がオンである	出力 O1 がオンである	入力 U/D に 24V 信号が入力 (カウントアップ)	入力クロックに信号が入力

制御バイト内の各ビットの設定によって、下記のような機能の実行が可能になります。

カウンタの設定: カウンタの初期値を出力チャンネル内の 32 ビットのカウンタ値①～④に設定し、制御バイトの Bit5 を設定します。このとき入力チャンネル内のカウンタ値①～④にこの初期値が書き込まれます。ビットを設定している限り、カウンタの動作は停止して、データは保存されます。カウントを開始するときは Bit5 を（Bit4 と共に）0 にします。

カウント動作：クロック入力にパルス信号が入力される毎に、入力チャンネル内のカウンタ値は上記初期値からカウントアップ（またはダウン）をします。出力チャンネル内の設定カウンタ値は変化しません。ステータスバイトの Bit0 はクロック入力が"high"のとき"1"になります。

カウンタの停止：制御バイトで Bit4 を設定すると、カウンタの処理動作を止めます。ステータスバイトの Bit4 がセットしていれば、カウンタ動作が止まったことを意味します。

カウンタのリセット：カウント動作後、制御バイト内 Bit5 を"1"にすると入力チャンネル内カウンタ値は出力チャンネル内カウンタ設定値に戻ります。

出力の設定：Bit 2 と 3 を使用して、カウンタモジュールに用意されている 2 つの出力をアクティブにするかどうかの設定を行います。カウンタが動作中でも操作可能です。

カウント結果はバイナリデータで表示されます。

例 (Ethernet、DeviceNet、CC-Link)

制御バイトにおいて“Bit5（カウンタの設定）を 1 にしてカウンタを設定します。

カウンタ初期値を 0x00123456（16 進数、4 バイト）とすると出力チャンネルは以下のようになります（0x□□は 16 進数、1 バイト）。

出力ワード 1		出力ワード 2		出力ワード 3	
0x00	0x20	0x34	0x56	0x00	0x12

このとき、入力チャンネルは以下のようになります。

入力ワード 1		入力ワード 2		入力ワード 3	
0x00	0x22	0x34	0x56	0x00	0x12

制御バイトにおいて“Bit5 を 0 にしてカウンタの開始準備をします。出力チャンネルは以下のようになります。

出力ワード 1		出力ワード 2		出力ワード 3	
0x00	0x00	0x34	0x56	0x00	0x12

クロック入力にパルスを入力すると、カウンタはパルス数に応じ増加（または減少）し、入力チャンネルは以下のようになります（増加の場合）。

入力ワード 1		入力ワード 2		入力ワード 3	
0x00	0x02	0x35	0x14	0x00	0x12

カウンタがカウント最大値に達した場合入力チャンネルは以下のようになります。

入力ワード 1		入力ワード 2		入力ワード 3	
0x00	0x02	0xFF	0xFF	0xFF	0xFF

さらにパルスが入力された場合はカウンタ値は一旦 0x00000000 になり、再びその値を増加します。カウンタ初期値に戻ります。

出力ワード 1		出力ワード 2		出力ワード 3	
0x00	0x20	0x34	0x56	0x00	0x12

このとき、入力チャンネルは以下のようになります。

入力ワード 1		入力ワード 2		入力ワード 3	
0x00	0x22	0x34	0x56	0x00	0x12

DLL を使用したときの制御例

① まずカウンタを停止させ、カウンタ値をリセットします。カウンタ値を具体的に設定する場合には制御バイト以外の３２ビットで設定します。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ値設定、リセット	カウンタ停止	出力 O2 をアクティブ（Hi）にする	出力 O1 をアクティブ（Hi）にする	0 不使用	0 不使用
0	0	1	1	0	0	0	0

＜ＶＢ上の設定値＞

ＤＬＬを利用して Visual　Basic でプログラムを組む場合の設定値：

dtData.nCtrl(1)　の値は

$2^5 + 2^4 = 48$　となります。

具体的にカウンタ値を設定したい場合は　dtData.nData(1)　で設定します。

－２１４７４８３６４８　～　＋２１４７４８３６４７　までの１０進数で設定できます。

② カウンタ動作は Bit5 、Bit4 をはずす（０に設定する）時点からスタートします。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ値設定、リセット	カウンタ停止	出力 O2 をアクティブ（Hi）にする	出力 O1 をアクティブ（Hi）にする	0 不使用	0 不使用
0	0	0	0	0	0	0	0

＜ＶＢ上の設定値＞

ＤＬＬを利用して Visual　Basic でプログラムを組む場合の設定値：

dtData.nCtrl(1)　の値は

0　となります。

③例えば２つの出力を High にするには Bit2 と Bit3 を「１」に設定します。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ値設定、リセット	カウンタ停止	出力 O2 をアクティブ（Hi）にする	出力 O1 をアクティブ（Hi）にする	0 不使用	0 不使用
0	0	0	0	1	1	0	0

＜ＶＢ上の設定値＞

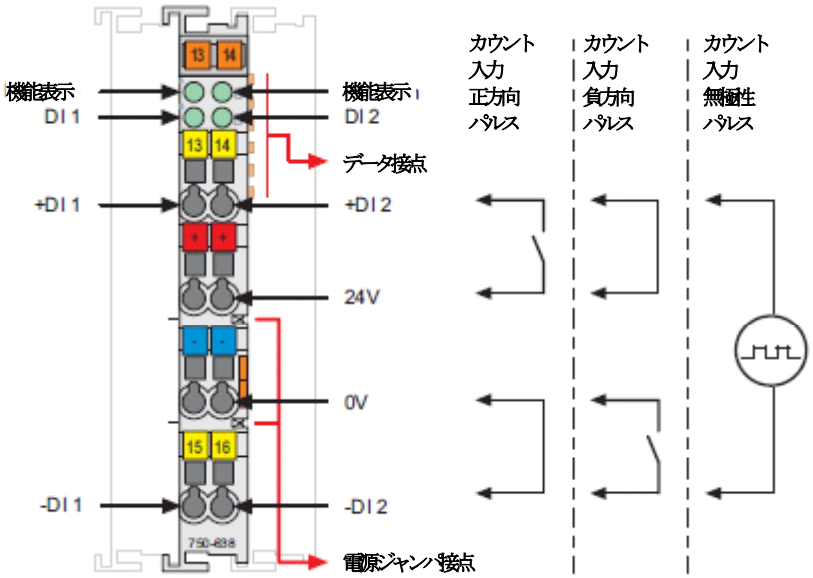
ＤＬＬを利用して Visual　Basic でプログラムを組む場合の設定値：

dtData.nCtrl(1)　の値は

$2^3 + 2^2 = 12$　となります。

注意 : 750-312/314/315/316 バスカプラを使用している場合は　DLL でカウンタを制御することは出来ません。（未対応）

10.2 カウンタモジュール
型番 750-638　2ch　アップ／ダウンカウンター　最大 500Hz



技術説明

このカウンタモジュールは差動入力信号の 24V 幅クロックのパルスをカウントして積算します。16 ビットのカウンタ 2 個を内臓していますので 2 種類のカウント入力を各々独立して扱うことができます。対応周波数は最大 500Hz です。

24V 巾のパルス信号を各入力チャンネル-DI 1 と+DI1 間、および-DI2 と+DI2 間に入力する場合、信号の電位は問いません。
一方、このモジュールにグランドを接続した場合は正方向または負方向のパルス入力をカウントすることができます。正方向信号（24V）は+DI1 または+DI2 に、負方向信号（0V）は-DI 1 または-DI2 に入力します。

シーケンサ（ホスト側）からは制御バイトによってカウンタのリセット、停止／スタート、カウンタ数値設定を行います。制御バイトをコントロールすることによってカウントアップとカウントダウンのコントロールも可能です。

注意：

このカウンターモジュールはシリアル通信バスカプラ（750-812,814,815,816）と Ethernet 通信バスカプラ（750-342, 750-842）ではプロセスイメージの大きさが異なりますのでご注意ください。

- **シリアル通信バスカプラ（750-812,814,815,816）の場合：４８ビット**
- **Ethernet 通信バスカプラ（750-342, 750-842）　の場合：６４ビット**

このモジュールには一番下の部分の電源ジャンパー接点が用意されていません。したがって、3 個全部の電源ジャンパー接点が必要なモジュール（例：2 チャンネルのデジタル入力モジュール）は、このモジュールの右側に直接接続することはできません。

技術データ

型番　750-	638
入力数	2
入力フィルタ	2ms
入力形式	**差動方式**
消費電流（内部ｋバスより）	10mA（平均）
駆動電圧	24V DC（-15%/+20%）
信号電圧（0）	-3V～+5V DC
信号電圧（1）	+15V～+30V DC
最大応答周波数	500Hz
入力電流（フィールド側）	**最大** 15mA
絶縁耐圧	500V（システムー電源　間）
内部ビット幅	48 ビット（シリアル通信バスカプラの場合） 64 ビット（Ethernet 通信バスカプラの場合）
動作温度範囲	0°C～+55°C
電線接続	**ケージクランプ；**0.08～2.5mm^2
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）

入力および出力データの構成 (750-812,750-814, 750-815, 750816) の場合

このカウンタモジュールは差動入力された２４Ｖ のクロックのパルスをカウントして積算します。１６ビットのカウンター２個を内臓しています。対応周波数は最大５００Hz です。

このモジュールは出力および入力のためにそれぞれ６バイト（４８ﾋﾞｯﾄ、３ワード分）のバッファを持っています。
各々のチャンネルに制御（ステータス）用に１バイト、１６ビットカウンター用に２バイト、が割り当てられています。

カウンター動作のストップ、スタート、カウンター値のセット、アップおよびダウンカウントの設定はすべて制御バイトにて行います。

２チャンネル分の制御バイトは同時に書き込んでください。片一方のチャンネル用の制御バイトだけを操作することはできません。

＜カウンタ値の設定＞

出力チャンネル下位		出力チャンネル中位		出力チャンネル上位	
制御バイト Ch1	Ch1 カウンタ値 1～8 ﾋﾞｯﾄ	Ch1 カウンタ値 9～16 ﾋﾞｯﾄ	制御バイト Ch2	Ch2 カウンタ値 1 ～ 8 ﾋﾞｯﾄ	Ch2 カウンタ値 9～16 ﾋﾞｯﾄ

＜カウンタ値の読出し＞

出力チャンネル下位		出力チャンネル中位		出力チャンネル上位	
ステータス バイト　Ch1	Ch1 カウンタ値 1～8ﾋﾞｯﾄ	Ch1 カウンタ値 9～16ﾋﾞｯﾄ	ステータス バイト　Ch2	Ch2 カウンタ値 1～8ﾋﾞｯﾄ	Ch2 カウンタ値 9～16ﾋﾞｯﾄ

出力データ中の制御バイトは下記の各ビットで構成されます。（Ch1, Ch2 共通）

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 **不使用**	0 **不使用**	**設定、リセット**(1) **動作開始**(0)	**ｶｳﾝﾀｰ停止**(1) **動作開始**(0)	**0** **不使用**	0 **不使用**	**ﾀﾞｳﾝｶｳﾝﾄ**(1) **ｱｯﾌﾟｶｳﾝﾄ**(0)	**ｵｰﾊﾞｰﾌﾛｰでｽﾄｯﾌﾟ**(1) **ｵｰﾊﾞｰﾌﾛｰでも動作**(0)

入力データ中のステータスバイトは下記の各ビットで構成されます。（Ch1, Ch2 共通）

ステータスバイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 **不使用**	**0** **不使用**	**値をセット中**(1) **動作中**(0)	**停止中**(1) **動作中**(0)	**0** **不使用**	**0** **不使用**	**ﾀﾞｳﾝｶｳﾝﾄ**(1) **ｱｯﾌﾟｶｳﾝﾄ**(0)	**ｵｰﾊﾞｰﾌﾛｰでｽﾄｯﾌﾟ**(1) **ｵｰﾊﾞｰﾌﾛｰでも動作**(0)

- 制御バイトの Bit5 でカウンター値を設定します。このﾋﾞｯﾄを設定している限りカウンター動作停止し、カウンタデータ値を直接書き込みできます。
- 制御バイトの Bit5 および Bit4 の設定をはずす（0 にする）とカウントを開始します。
- カウンターがオーバーフローしても動作を継続したい場合は制御バイトの Bit０を（０）に設定します。
- **２チャンネル分の制御バイトは同時に書き込んでください。片一方のチャンネル用の制御バイトだけを操作することはできません。**

カウンターモジュールの制御例 (750-812, 750-814, 750-815, 750-816 の場合)

① まずカウンターを停止させ、カウンター値をリセットします。カウンター値を具体的に設定する場合には制御
バイト以外の１６ビットで各々のチャンネルに値を設定します。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	設定、リセット(1)	ｶｳﾝﾀｰ停止 (1)	0	0	ﾀﾞｳﾝｶｳﾝﾄ(1) ｱｯﾌﾟｶｳﾝﾄ(0)	ｵｰﾊﾞｰﾌﾛｰでｽﾄｯﾌﾟ (1) ｵｰﾊﾞｰﾌﾛｰでも動作 (0)

＜ＶＢ上の設定値＞

ＤＬＬを利用して Visual　Basic でプログラムを組む場合の設定値：
アップカウント、オーバフローでも動作という条件で設定すると、
dtData.nCtrl(1)、dtData.nCtrl(2)　　の値は
$2^5 + 2^4 = 48$　となります。
具体的にカウンタ値を設定したい場合は　各々のチャンネルに
０　～　＋６５５３５　までの１０進数で設定できます。

② カウンター動作は Bit5 、Bit4 をはずす（０に設定する）時点からスタートします。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ動作 開始	カウンター動作 開始	不使用	不使用	0 ｱｯﾌﾟｶｳﾝﾄ	0 ｵｰﾊﾞｰﾌﾛｰ許可
0	0	0	0	0	0	0	0

＜ＶＢ上の設定値＞

ＤＬＬを利用して Visual　Basic でプログラムを組む場合の設定値：
dtData.nCtrl(1)　dtData.nCtrl(2)の値は
０　となります。

③例えばカウンターをダウンカウントに設定するには Bit1 を(1)にします。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	カウンタ動作 開始	カウンター動作 開始	不使用	不使用	1 ﾀﾞｳﾝｶｳﾝﾄ	0 ｵｰﾊﾞｰﾌﾛｰ許可
0	0	0	0	0	0	1	0

＜ＶＢ上の設定値＞

ＤＬＬを利用して Visual　Basic でプログラムを組む場合の設定値：
dtData.nCtrl(1) 、dtData.nCtrl(2)　の値は
$2^1 = 2$　となります。

注意：750-312/314/315/316 バスカプラを使用している場合は　DLL でカウンターを制御することは出来ません。（未対応）

入力および出力データの構成 (750-342, 750-842 の場合)

このカウンタモジュールは差動入力された２４Ｖ のクロックのパルスをカウントして積算します。１６ビットのカウンタ２個を内臓しています。対応周波数は最大５００Hz です。

このモジュールは出力および入力のためにそれぞれ８バイト（６４ﾋﾞｯﾄ、２ch 分）のバッファを持っています。
各々のチャンネルに制御（ステータス）用に１バイト、１６ビットカウンタ用に２バイト、が割り当てられています。

カウンタ動作のストップ、スタート、カウンタ値のセット、アップおよびダウンカウントの設定はすべて制御バイトにて行います。
750-342 バスカプラ／750-842 コントローラ内の 750-638 のプロセスイメージは以下のとおりです。

出力領域	上位バイト	下位バイト
ワード１	空き	Ch1 制御バイト
ワード２	Ch1 カウンタ　9～16 ビット	Ch1 カウンタ　1～8 ビット
ワード３	空き	Ch2 制御バイト
ワード４	Ch2 カウンタ　9～16 ビット	Ch2 カウンタ　1～8 ビット

入力領域	上位バイト	下位バイト
ワード１	空き	Ch1 ステータスバイト
ワード２	Ch1 カウンタ　9～16 ビット	Ch1 カウンタ　1～8 ビット
ワード３	空き	Ch2 ステータスバイト
ワード４	Ch2 カウンタ　9～16 ビット	Ch2 カウンタ　1～8 ビット

Modbus/TCP マスタ上のデータフォマットは以下のフォーマットになります。

出力データ（カウンタ値の設定）

出力データ	上位バイト	下位バイト
出力ワード１	不使用	Ch1 制御バイト
出力ワード２	Ch1 カウンタ　9～16 ビット	Ch1 カウンタ　1～8 ビット
出力ワード３	不使用	Ch2 制御バイト
出力ワード４	Ch2 カウンタ　9～16 ビット	Ch2 カウンタ　1～8 ビット

入力データ（カウンタ値の読出し）

入力データ	上位バイト	下位バイト
入力ワード１	不使用	Ch1 ステータスバイト
入力ワード２	Ch1 カウンタ　9～16 ビット	Ch1 カウンタ　1～8 ビット
入力ワード３	不使用	Ch2 ステータスバイト
入力ワード４	Ch2 カウンタ　9～16 ビット	Ch2 カウンタ　1～8 ビット

カウンタ動作のストップ、スタート、カウンタ値のセット、アップおよびダウンカウントの設定はすべて制御バイトにて行います。

出力データ中の制御バイトは下記の各ビットで構成されます。（Ch1, Ch2 共通）

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	設定、ﾘｾｯﾄ(1) 動作開始(0)	ｶｳﾝﾀｰ停止(1) 動作開始(0)	0 不使用	0 不使用	ﾀﾞｳﾝｶｳﾝﾄ(1) ｱｯﾌﾟｶｳﾝﾄ(0)	ｵｰﾊﾞｰﾌﾛｰでｽﾄｯﾌﾟ(1) ｵｰﾊﾞｰﾌﾛｰでも動作(0)

入力データ中のステータスバイトは下記の各ビットで構成されます。（Ch1, Ch2 共通）

ステータスバイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 不使用	0 不使用	値をｾｯﾄ中(1) 動作中(0)	停止中(1) 動作中(0)	0 不使用	0 不使用	ﾀﾞｳﾝｶｳﾝﾄ(1) ｱｯﾌﾟｶｳﾝﾄ(0)	ｵｰﾊﾞｰﾌﾛｰでｽﾄｯﾌﾟ(1) ｵｰﾊﾞｰﾌﾛｰでも動作(0)

カウンタの設定：制御バイトの Bit5 でカウンター値を設定します。このﾋﾞｯﾄを設定している限りカウンター動作は停止し、カウンタデータ値を直接書き込みできます。
カウンターがオーバーフローしても動作を継続したい場合は制御バイトの Bit 0 を（0）に設定します。

カウンタの開始：制御バイトの Bit5 および Bit4 の設定をはずす（0 にする）とカウントを開始します。

カウンターモジュールの制御例 (750-342、750-842 の場合)

① まずカウンターを停止させ、カウンター値をリセットします。カウンター値を具体的に設定する場合には制御バイト以外の１６ビットで各々のチャンネルに値を設定します。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	1 設定、ﾘｾｯﾄ	1 ｶｳﾝﾀｰ停止	0	0	0 ｱｯﾌﾟｶｳﾝﾄ	0 ｵｰﾊﾞｰﾌﾛｰでも動作

具体的にカウンタ値を設定したい場合は各々のチャンネルに 0～＋65535 までの１０進数で設定できます。

② カウンター動作は Bit5 、Bit4 をはずす（0 に設定する）時点からスタートします。

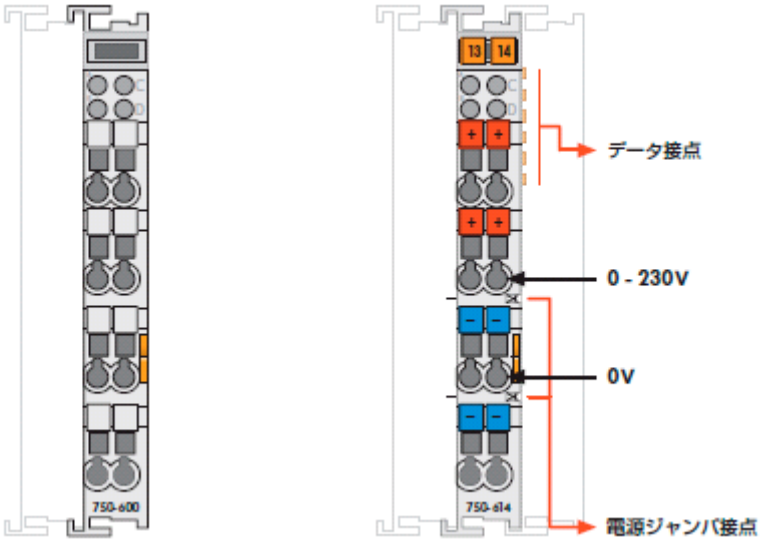
制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	0	0

③ 例えばカウンターをダウンカウントに設定するには Bit1 を(1)にします。

制御バイト							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	0	0	1 ﾀﾞｳﾝｶｳﾝﾄ	0 ｵｰﾊﾞｰﾌﾛｰでも動作

11 その他のモジュール

11.1 終端モジュール、電源拡張モジュール 型番 750-600, 750-614



技術説明

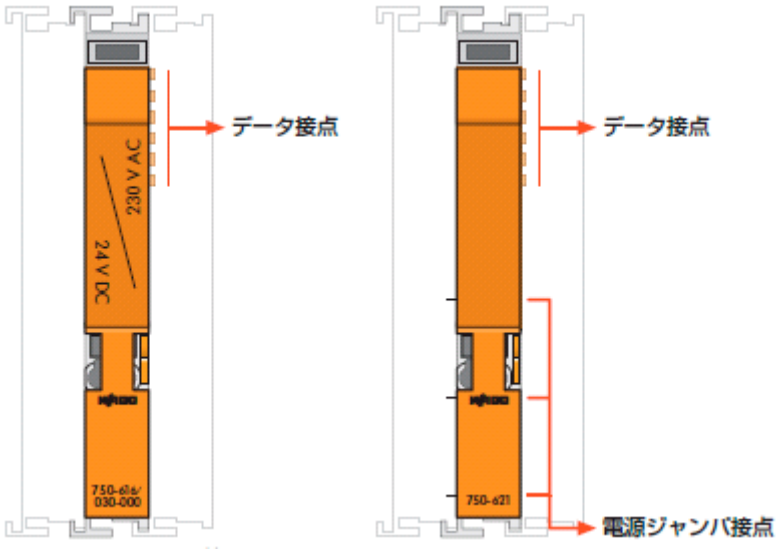
750-600 はバスカプラと必要な I/O モジュールを使用してフィールドバス・ノードのアセンブリが完了した時点で、最右端に設置するモジュールです。この作業によって内部データ回路がすべて完成したことになり、内部バス（K-bus）のデータフローが保証されます。

750-614 電源拡張モジュールを使用すると、正と負のコモン端子をさらに追加することが可能です（最高各 4 点まで）。750-402、750-504 などの 4 チャンネルモジュールを使用した場合隣に挿入すると有効です。したがって、端子台を外付けする必要がありません。

技術データ

型番　750-	600	614
電圧	－	24V～230V AC/DC
接点上の電圧	－	10mA（最大）
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm²	
外形寸法（mm）　W×H×L	12×64×100（キャリアレールの上端から測定）	

11.2 セパレーションモジュール
型番 750-616, 750-616/030-000, 750-621



技術説明

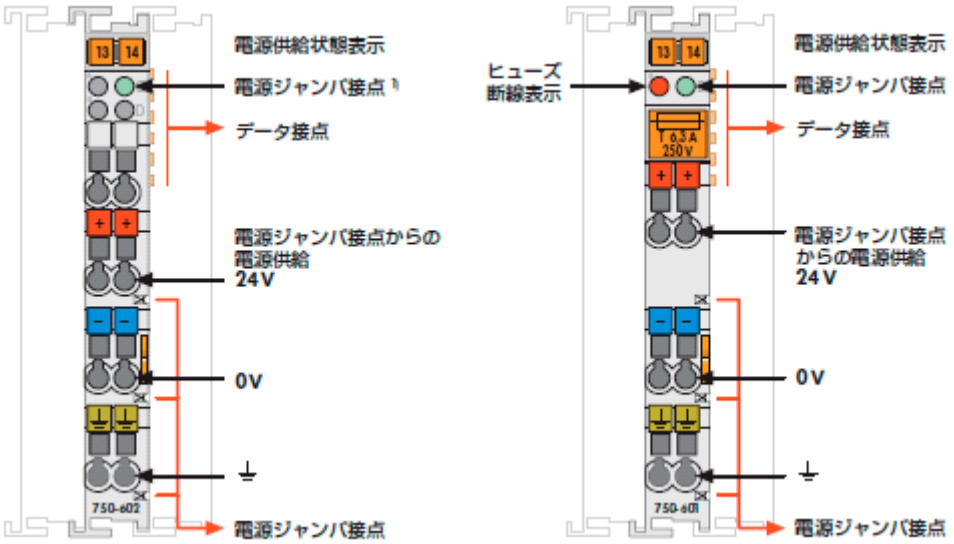
このモジュールを使用すると、ノード内部の異なるフィールド電圧間の空間および沿面距離を広げることが可能です。

型番 750-616 は上図の左側において電圧表示のないタイプです。型番 750-616/030-000 は上図の左側に示すタイプです。
型番 750-621 には電源ジャンパ接点があり、このモジュール以降に対し、フィールド電源を供給することができます。

技術データ

型番	750-616, 750-616/030-000, 750-621
外形寸法（mm）　W×H×L	12×64*×100（***キャリアレールの上端から測定**）

11.3 電源入力モジュール
型番 750-601, 602, 609, 610, 611, 612, 613, 615



技術説明

電源入力モジュールは、電源ジャンパー接点を通してセンサーなどが接続された I/O モジュールに外部機器用の電源を供給します。すべての接続モジュールに供給される電源電流の最大値は 10A です。ヒューズホールダが取付けられたモジュールへの電源電流の最大値は 6.3A です。これよりも高い電流が必要な場合には、中間の電源入力モジュールをアセンブリに追加実装することが可能です。
モジュール 750-601, 609, 615, 610 および 611 にはヒューズホールダが追加で取付けられています。ヒューズの交換は非常に簡単で、ヒューズホールダを引き出してヒューズを交換するだけです。ヒューズの断は LED で表示されます。
モジュール 750-610 および 611 は、電源入力モジュールのステータスに関する情報を 2 つの入力ビットで用意します。フィールドバスカプラからこれを読み出すことができます。

ビット 1	**ビット** 2	**説明**
0	0	**電圧<**15V DC
1	0	**ヒューズ断**
0	1	**ヒューズ** OK、**電圧** OK

電源入力モジュールの使用に際しては、その許容電源電圧について注意を払うことが必要です。電源入力モジュールに規定されている定格電圧を次ページの表に記載しています。

電源入力モジュール 750-613 は、フィールド側と内部データバスシステム用の両方に電源を供給します。内部システム電源電流の最大値は 2A として規定されています。内部消費電流の合計がこの 2A を超える場合には、本電源入力モジュールの追加が必要になります。

技術データ

型番　750-	602	612	613
電圧	24V DC	0〜230V AC/DC	24V DC (-15%/+20%)
接点経由の電流	10A（最大）		
動作温度範囲	0°C〜+55°C		
電線接続	ケージクランプ；0.08〜2.5mm²		
外形寸法（mm）　W×H×L	12×64×100（キャリアレールの上端から測定）		

モジュール 750-613 の内部電流：2A（最大値）

型番　750-	601	609	615
電圧	24V DC	230V AC	120V AC
接点経由の電流 **ヒューズ**	6.3A（最大） 5×20, 6.3A		
動作温度範囲	0°C〜+55°C		
電線接続	ケージクランプ；0.08〜2.5mm²		
外形寸法（mm）　W×H×L	12×64×100（キャリアレールの上端から測定）		

型番　750-	610	611
入力数	2	
消費電流	5mA	
内部ビット幅	2	
電圧	24V DC	230V AC
接点経由の電流 **ヒューズ**	6.3A（最大） 5×20, 6.3A	
動作温度範囲	0°C〜+55°C	
電線接続	ケージクランプ；0.08〜2.5mm²	
外形寸法（mm）　W×H×L	12×64×100（キャリアレールの上端から測定）	

11.4 シリアルインタフェースモジュール

RS232C インタフェース

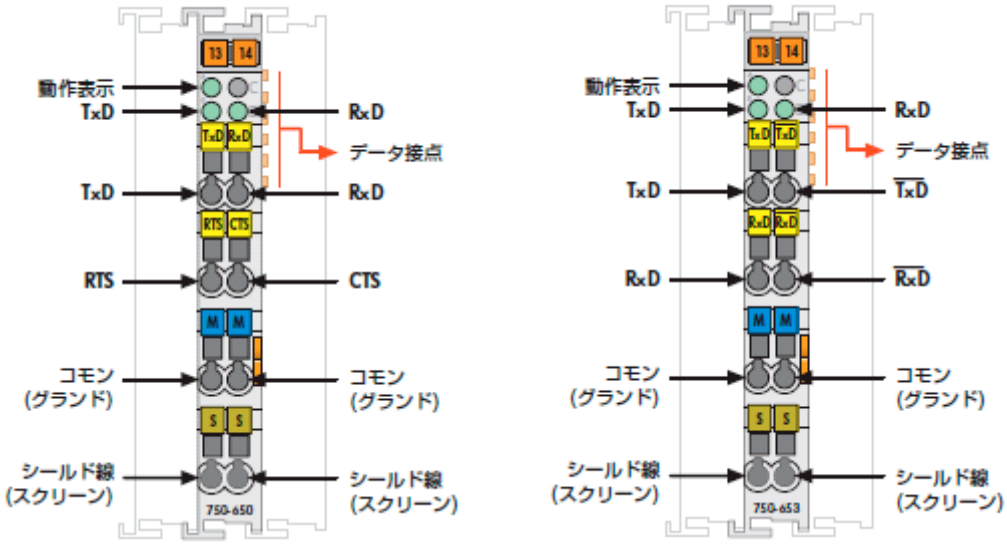
型番　750-650（9600bps、ﾊﾟﾘﾃｨ無、8ﾋﾞｯﾄ、ｽﾄｯﾌﾟﾋﾞｯﾄ 1）

750-650/003-000（通信条件可変タイプ）

RS485 インタフェース

型番　750-653（9600bps、ﾊﾟﾘﾃｨ無、8ﾋﾞｯﾄ、ｽﾄｯﾌﾟﾋﾞｯﾄ 1）

750-653/003-000（通信条件可変タイプ）



技術説明

RS232C インタフェース標準モジュール 750-650 および RS485 インタフェース標準モジュール 750-653 は機能設定が固定であり、出荷時に通信速度 9600 ボー、パリティビット無、データ長 8 ビット、ストップビット 1 に設定されています。これ以外の各通信パラメータの組合せは、注文時に型番に対し枝番を指定する（W4-Vol3 カタログ参照）ことによって入手することができます。

一方、工場出荷後各パラメータを設定できる通信条件可変タイプとして RS232C 用には 750-650/003-000、RS485 用には 750-653/003-000 があります。これらの可変タイプも工場出荷時には通信速度 9600 ボー、パリティビット無、データ長 8 ビット、ストップビット 1 に設定されています。この通信条件を変更・設定するためには WAGO-I/O-CHECK 2 設定ツールを用います。

技術データ

シリーズ 750-	650 650/003-000	653 653/003-000
伝送チャンネル	全二重（フロー制御）	全二重、半二重
伝送速度	1200～19200 ボー	
ビットスキュー	＜3%	－
ビット伝送	－	ISO 8482/DIN 66259 T4 に適合
負荷抵抗	－	－
消費電流（内部）	最大 55mA	最大 65mA
伝送ケーブル長	最大長が 15m の RS232 ケーブル	最大長が 500m の ツイストペア線
入力バッファ	128 バイト	
出力バッファ	16 バイト	
電源電圧	内部システム電源経由で供給	
絶縁耐圧	500V システム／電源	
内部ビット幅	1×最大 40 ビット、1×8 ビットの制御／ステータス	
動作温度範囲	0°C～+55°C	
電線接続	ケージクランプ；0.08～2.5mm²	
電線むき長さ	8-9mm	
外形寸法（mm）　W×H×L	12×64*×100（*キャリアーレールの上端から測定）	
出荷時にプリセットされている通信条件	750-650 750-653	9600 ボー、パリティ無、データ長 8 ビット ストップビット 1
	750-650/003-000 750-653/003-000	通信条件可変 WAGO-I/O-CHECK 2 により設定

RS232 の説明

この RS232 インタフェースモジュールは、すべてのワゴ I/O フィールドバスカプラと共に使用することができます。本インタフェースモジュールを用いると、RS232 インタフェースを持った機器は WAGO-I/O-SYSTEM に直接接続することが可能です。このインタフェースモジュールは、フィールドバス通信内で RS232 機器からのデータをフィールドバスに渡すゲートウェイ機能を提供します。すなわち、プリンタ、バーコードリーダー、そしてローカルオペレータインタフェースの通信リンクなどのシリアル装置は、フィールドバスカプラ／コントローラを介して PLC または PC マスタと直接通信をすることができます。

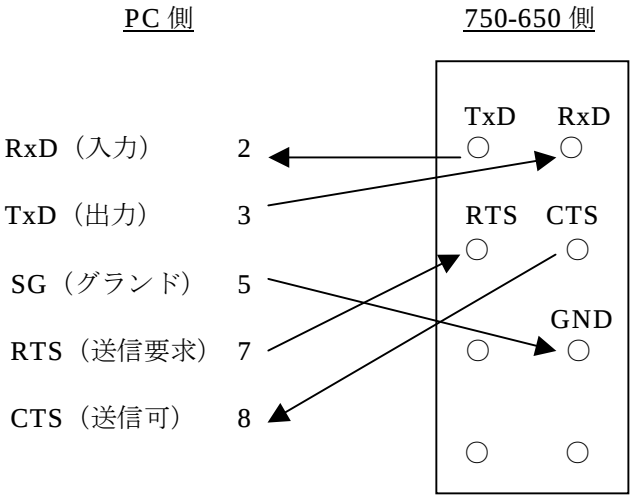
通信動作は上位フィールドバスに対して独立して行われます。このインタフェースモジュールは上位レベルの通信プロトコルを持っていませんので、プロトコルの構築はマスタ PLC あるいは PC 側で行うことになります。データ伝送を保証するために、128 バイトの受信バッファと 16 バイトの送信バッファが用意されています。

データ通信は RTS と CTS を使用してハンドシェイクを行い、フロー制御にて行うことが可能ですが、この機能を使わないとき（相手方の装置にハンドシェイク機能が無いとき）は RTS と CTS は短絡しておきます。装置によってはハンドシェイク機能を持たないもので、RTS と CTS の結線を必要とするものもありますので注意が必要です。

標準のデータ伝送は 9,600 ボーレートの速度で行われます。1 スタートビット、8 データビットおよび 1 ストップビットが送信されます。パリティビットは用意されていません。
オプション仕様では最大 19,200 ボーまでの全二重通信が可能となります。

モジュール上部には動作状態を示す表示用 LED（動作中、TxD、RxD）が用意されており、動作が正常に行われているかどうかを確認することができます。

例：PC と RS232C モジュールの接続
PC 側のシリアルポート（DOS-V 9 ピン）と 750-650 を接続する場合のピンアサインは以下のようになります。



PC 側の設定：

転送速度	：9600 ビット／秒
データビット	：8 ビット
パリティ	：なし
ストップビット	：1
フロー制御	：なし

RS485 の説明

この RS485 インタフェースモジュールは、すべてのワゴ I/O フィールドバスカプラと共に使用することができます。本インタフェースモジュールを用いると、RS485 インタフェースを持った機器は WAGO-I/O-SYSTEM に直接接続することが可能です。このインタフェースモジュールは、フィールドバス通信内で RS485 機器からのデータをフィールドバスに渡すゲートウェイ機能を提供します。すなわち、RS485 インタフェースを持ったシリアル装置は、フィールドバスカプラ／コントローラを介して PLC または PC マスタと直接通信をすることができます。

通信動作は上位フィールドバスに対して独立して行われます。このインタフェースモジュールは上位レベルの通信プロトコルを持っていませんので、プロトコルの構築はマスタ PLC あるいは PC 側で行うことになります。データ伝送を保証するために、128 バイトの受信バッファと 16 バイトの送信バッファが用意されています。

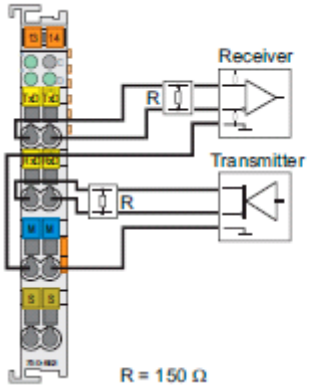
RS485 においては RS232C のようにハードウェアによるフロー制御機能がありませんので、マルチドロップ接続で複数の機器と通信する場合には、各々の機器との通信手順をシーケンサあるいは WAGO-IO-PRO 側でプログラムする必要があります。

標準のデータ伝送は 9,600 ボーレートの速度で行われます。1 スタートビット、8 データビットおよび 1 ストップビットが送信されます。パリティビットは用意されていません。
モジュールの拡張機能としては全二重通信で 19,200 ボーまでをサポートします。
一方、バス接続で半二重通信を行うときは送信と受信を同時に実行することはできません。

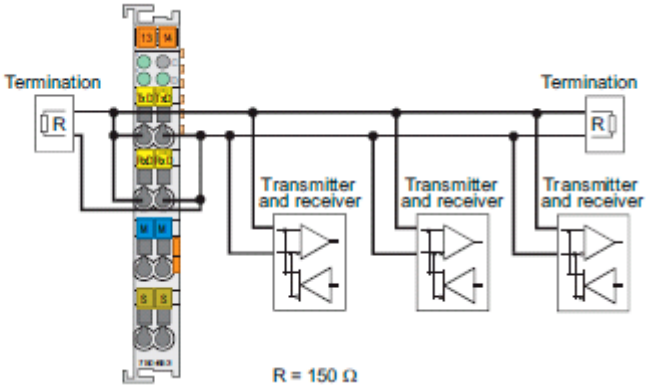
RS485 通信機器との接続は、TxD、/TxD、RxD、/RxD およびグランドの各端子を用いて行われます。ケーブルのシールド線はシールド端子に接続することにより、直接 DIN レールにつながります。

モジュール上部には動作状態を示す表示用 LED（動作中、TxD、RxD）が用意されており、動作が正常に行われているかどうかを確認することができます。

RS485 機器との接続は以下に示すようにポイントツーポイント接続およびバス接続の 2 つの方式があります。接続ライン上には図のように 150 Ω の終端抵抗が必要になります。



ポイントツーポイント接続



バス接続

入力および出力データの構成

このモジュールは、標準で 2 ワードの入力／出力領域を持っており、複合の入力／出力モジュールと同様なバスモジュールと考えられます。送受信データの転送は、標準で 3 バイトまでの出力と 3 バイトまでの入力の各バイトデータを使用して行われます。入力／出力領域はオプションで 3 ワードまで拡張が可能です（入力／出力データは各 5 バイトまで拡張可能）。フローティングデータの制御には、1 つの制御バイトと 1 つのステータスバイトをそれぞれ使用します。通信上のフロー制御は、制御バイト内の Bit1 または Bit0 をトグルビットとして用いることにより行われます。

モジュール内のプロセスイメージは以下のとおりです。

出力部：

	上位バイト	下位バイト
ワード 0	出力バイト 0（D0）	制御バイト（C）
ワード 1	出力バイト 2（D2）	出力バイト 1（D1）
ワード 3（オプション）	出力バイト 4（D4）	出力バイト 3（D3）

入力部：

	上位バイト	下位バイト
ワード 0	入力バイト 0（D0）	ステータスバイト（S）
ワード 1	入力バイト 2（D2）	入力バイト 1（D1）
ワード 3（オプション）	入力バイト 4（D4）	入力バイト 3（D3）

標準では受信したデータのうち 1 回の取り込みで、入力データとして 3 バイト分が入力バイト 0～2 に入ります。一方、出力バイト 0～2 の中には送信されるデータが格納されます。

出力データ中の制御バイトは下記の各ビットで構成されます。

制御バイト（C）							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	OL2	OL1	OL0	0	IR	RA	TR
常に 0	送受信するバイト数を標準では OL1、OL0 の組合せ（0～3）で、拡張版では OL2、OL1、OL0 の組合せ（0～5）で設定する。			常に 0	シーケンスの初期化	受信要求および受信データの取り込みのトグルビット	データ送信するためのトグルビット

入力データ中のステータスバイトは下記の各ビットで構成されます。

ステータスバイト（S）							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	IL2	IL1	IL0	BUF_F	IA	RR	TA
常に 0 であることが必要	データを送受信したときに制御バイトに設定した OL2～OL0 のビット値に対応して応答メッセージ中で IL2～IL0 の各ビットがセットされる。			受信バッファの容量が満杯のとき 1	初期化コマンドで初期化された直後にこのビットが立つ	データの受信要求、データ読み込みのための確認ビット	データ送信トグルビットの確認

制御バイトとステータスバイトを使用して、以下のようにデータの送受信を制御することができます。

モジュールの初期化：
モジュールの初期化を行うには、制御バイトの IR（Bit2）を"1"にセットします。この設定を行うと、データの送受信機能は停止します。送受信データはすべて消去されます。送信バッファの値は初期化後もレジスタにそのまま残ります。

	1 ワード目		2 ワード目	
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00（D0）	04（C）	00（D2）	00（D1）
	3 ワード目（オプション）			
2 進数	0000 0000	0000 0000		
16 進数	00（D4）	00（D3）		

データの送信：
送信データの数量は OL0～OL2 で表されたバイト数によって示されます。標準では 3 バイトづつ送信され、オプションで 5 バイトまで設定が可能です。制御バイト中の TR（Bit0）の現在値は送信毎に変える必要があります（1 と 0 を交互にセット）。TR に新しいビット値が書き込まれた後ステータスバイトを読み、TA に同じ値がセットされたことを確認します。この後、送信バッファへのデータ転送が実行されます。TR が前回と変化していない場合は、書き込まれたデータが新しいデータとして認識されませんので送信バッファへのデータ転送は行われません。

以下の送受信の例は PLC 側で割当てられるワード内のデータの様子です。

データ送信方法（標準の場合）：

1）最初に、モジュールを初期化するためにデータを"0"にして制御バイトの IR（Bit2）を"1"にセットします。この設定を行うと、データの送受信は停止します。受信バッファのデータはすべてクリアされます。

	1 ワード目		2 ワード目	
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00	04	00	00

2）次に制御バイトの TR を"1"にして 3 バイト分のデータを送ります。

2 進数	XXXX XXXX	0011 0001	XXXX XXXX	XXXX XXXX
16 進数	□□（D0）	31	□□（D2）	□□（D1）

－データはモジュールから、D0、D1、D2 の順で出力されます。
－この後ステータスバイトを読むと TA=1 になっています。

3）次の 3 バイトを送るには制御バイトの TR を"0"にして書き込みます。

2 進数	XXXX XXXX	0011 0000	XXXX XXXX	XXXX XXXX
16 進数	△△（D0）	30	△△（D2）	△△（D1）

－データはモジュールから、D0、D1、D2 の順で出力されます。
－この後ステータスバイトを読むと TA=0 になっています。

4）以降これを繰り返します。

データの受信：
制御バイト中の RA をトグルビットとして用い、ステータスバイト中の RR で同じ値がセットされたかどうかを確認します。最初に初期化された後、データ受信の準備のために RA を"1"にセットする必要があります。データを受信したら制御バイトの RA を"0"にすると OL0～OL2 で表されたバイト数（標準＝3 バイト）のデータを受信バッファから読み込みます。データの読み込みが終了すると、RR が符号反転されます。これを

確認した後 RA を反転し、次のデータを読み込みます。これを繰り返します。

データの送信と受信は同時に行うことが可能です。

データ受信方法（標準の場合）：

1）最初に、受信バッファを初期化するためにデータを"0"にして制御バイトで初期化します。

	1 ワード目		2 ワード目	
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00	04（C）	00	00

2）受信の準備のため RA を"1"にしてデータ受信 OK を知らせます。

2 進数	0000 0000	0000 0010	0000 0000	0000 0000
16 進数	00	02（C）	00	00

3）外部シリアル機器から文字データを 3 バイト分発生し、モジュールに入力します。データは受信バッファに入ります。

4）制御バイトの RA を"0"にして、3 バイト分のデータを受信バッファから取り込みます。

2 進数	0000 0000	0011 0000	0000 0000	0000 0000
16 進数	00	30（C）	00	00

－ここで、レジスタ読み込みコマンドでデータを読み込みます。このとき、ステータスビット RR は自動的に"1"に変化します。

2 進数	XXXX XXXX	0011 0010	XXXX XXXX	XXXX XXXX
16 進数	□□（D0）	32（S）	□□（D2）	□□（D1）

5）次の 3 バイトを取り込むためには制御バイトの RA を"1"にして受信バッファから取り込みます。

2 進数	0000 0000	0011 0010	0000 0000	0000 0000
16 進数	00	32（C）	00	00

－ここで、レジスタ読み込みコマンドでデータを読み込みます。このとき、ステータスビット RR は自動的に"0"になります。

2 進数	XXXX XXXX	0011 0000	XXXX XXXX	XXXX XXXX
16 進数	△△（D0）	30	△△（D2）	△△（D1）

以降、RA を交互にビット値を変えて制御バイトを送ることにより、データを 3 バイトづつ読み込みます。

＊ TR と RA を同時に操作しながら、データの送信と受信を同時に行うことが可能です。
この場合は、3 バイトと制御バイトを送った後、レジスタを読み込むという動作を繰り返します。

＊ 受信バッファの容量が満杯になるとステータスバイト中の **Buffer Full（Bit 3）**が"1"になります。この状態で外部から受信してもデータは取りこぼしの状態になります。従って、
初期化コマンドで受信バッファをクリアする必要があります。

データの送信例

データ文字列"Hello!"の送信

1）モジュールを初期化するために制御バイトの IR（Bit2）を"1"にセットします。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00	04	00	00

－この後ステータスバイトを読むと IA=1 になっています。

	入力バイト 0	ステータスバイト	入力バイト 2	入力バイト 1
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00	04	00	00

２）制御バイトの TR（Bit0）を"1"にして、3 バイト分のデータを送ります。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0100 1000	0011 0001	0110 1100	0110 0101
16 進数	48(H)	31	6C(l)	65(e)

３）制御バイトの TR（Bit0）を"0"にして、残りの 3 バイト分のデータを送ります。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0110 1100	0011 0000	0010 0001	0110 1111
16 進数	6C(l)	3０	21(!)	6F(o)

データの受信例

データ文字列"WAGO"の受信

１）モジュールを初期化するために制御バイトの IR（Bit2）を"1"にセットします。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00	04	00	00

－この後ステータスバイトを読むと IA=1 になっています。

	入力バイト 0	ステータスバイト	入力バイト 2	入力バイト 1
2 進数	0000 0000	0000 0100	0000 0000	0000 0000
16 進数	00	04	00	00

２）受信を可能にするため RA（Bit1）を"1"にセットします。受信の準備が完了します。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0000 0000	0000 0010	0000 0000	0000 0000
16 進数	00	02	00	00

－このときステータスバイトは IA=0 にリセットされます。

３）ここで、外部のシリアル通信機器（PC のハイパーターミナルなど）からデータを送信します。"WAGO" 文字列が受信バッファに入ります。

4）この後、制御バイトの RA（Bit1）を"0"にして、3 バイト分のデータを受信バッファから取り込みます。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0000 0000	0011 0000	0000 0000	0000 0000
16 進数	00	30	00	00

－ここで、レジスタ読み込みコマンドでデータを読み込みます。このとき、ステータスビット RR は自動的に"1"に変化します。

	入力バイト 0	ステータスバイト	入力バイト 2	入力バイト 1
2 進数	0101 0111	0011 0010	0100 0111	0100 0001
16 進数	57(W)	32	47(G)	41(A)

5）次の 3 バイトを取り込むためには制御バイトの RA を"1"にして受信バッファから取り込みます。

	出力バイト 0	制御バイト	出力バイト 2	出力バイト 1
2 進数	0000 0000	0011 0010	0000 0000	0000 0000
16 進数	00	32	00	00

－この後、レジスタ読み込みコマンドでデータを読み込みます。このとき、ステータスビット RR は自動的に"0"になります。

	入力バイト 0	ステータスバイト	入力バイト 2	入力バイト 1
2 進数	0100 1111	0011 0000	0000 0000	0000 0000
16 進数	4F(O)	30	00	00

注意：

PLC 側で制御プログラムを組む場合、フィールドバスの種類によって I/O のプロセスイメージと PLC 内のメモリマッピングが異なることがありますので注意してください。
DeviceNet、CC-Link、Ethernet の場合は PLC 内のメモリマッピングは前記プロセスイメージと同じ並び順になります。
一方、Profibus の場合は PLC 内のメモリマッピングは実装 I/O の並び順でバイト単位で配置されます。また、8 ビット以下のモジュールが先に実装され、後に 8 ビットモジュールが続いた場合は最初のバイトの上位 4 ビットは空きとなります。

12 LabVIEW 開発用ソフトウェアライブラリ

LabVIEW 開発システムを持っているユーザはこの章を参照することにより、ワゴ I/O システムの制御が可能になります。LabVIEW を持っていないが、LabVIEW により作成された実行ソフトウェアの取扱についてははご覧下さい。

12.1 ファイル構成

「シリアル通信ワゴ I/O システム」には以下の LabVIEW 用ソフトウェアが用意されております。これらは CD-ROM またはホームページ http://www.wago.com/wagoweb/japan/jap/index.htm）からのダウンロードによって提供されます。CD の場合はフォルダーProgram Files 以下をすべて PC のルート C:¥にコピーします。ダウンロードの場合は最初に vi_lib.exe（自己解凍形式）を選択し、実行します。解凍先ディレクトリが要求されますので適切な場所を指定し、解凍すると Program Files が作成されます。ここで Program Files¥以下には Wago_io¥WdllCall¥があり、この中に下記のファイルが収められていますが、この Program Files¥以下全てをルート C:¥にコピーします。

WdllCall.dll	基本 DLL である WAGO_IO.dll（第 6 章参照）を呼び出して、LabVIEW で使用できるようにデータ構造変換を行う DLL です。
WagoIO.llb	基本命令、目的別命令などが収められています（下記参照)。「DLL 呼び出し用 VI」および「WagoIO 入出力 VI」が入っています。
WAGO_DAQ**	標準添付データ収集ソフトウェアのソースダイアグラムです。
WagoSample.llb	目的別命令を使用して、データ収集などのサンプルが収められています（下記参照)。
Samples.vi	サンプルを起動するだけの VI です。
DAQ_Apply.vi	データ収集のアプリケーションです、収集したデータをログファイルに出力します。

12.1.1 WAGO_IO.dll 呼び出し用 DLL（WdllCall.dll）

この DLL は以下の各節（１２－２節以降）に記述されている各 VI 中で必要に応じて使用されています。LabVIEW のデータ構造に合わせてデータ変換をした上で WAGO_IO.dll 内の対応する各関数を呼び出しています。フォルダーProgramFiles¥Wago_io¥WdllCall¥の下に置かれています。

使用方法

WdllCall.dll を WAGO_IO.dll と同じフォルダ、もしくはパスの通ったフォルダに配置すること。
（LabVIEW の自動探査機能を使用すれば、どこに置いておいても構いません）
LabVIEW で、WdllCall.dll を呼び出す VI を開いた場合に対象となる DLL が見つからなかった場合は自動的に DLL を探しにいきます。
プログラムを作成するときは VI 編集画面において、DLL 呼び出し用の VI（１２－２節）をブロックダイアグラムに配置し、必要なパラメータを制御器や定数で引き渡す事で使用できます。

以下に WAGO_IO.dll を動作させるための呼び出し DLL の対応関数一覧表を掲げます。

	関数名 (WAGO_IO.dll 内)	機能・役割
通信動作	WDL_OpenCommPort (WAGO_OpenCommPort)	ポートをオープンするための関数
	WDL_CloseCommPort (WAGO_CloseCommPort)	ポートをクローズのための関数
ﾓｼﾞｭｰﾙ ｺﾝﾌｨｸﾞﾚｰｼｮﾝ	WDL_SetConfig (WAGO_SetConfig)	指定ノードのモジュール構成情報を DLL に通知するための関数
データの 入出力	WDL_GetData (WAGO_GetData)	入力モジュールのデータ読み込み処理関数
	WDL_SetDigitalData (WAGO_SetDigitalData)	ディジタル出力モジュールへのデータ出力関数
	WDL_SetAnalogData (WAGO_SetAnalogData)	アナログ出力モジュールへのデータ出力関数

12.1.2 WagoIO.llb 概要

基本命令と目的別命令の 2 系統があります。目的別命令からは基本命令を呼び出しています。

基本命令 (WAGO_IO.dll に準拠)

- SetConfig.vi　　ノードのコンフィグレーションを行います。
- OpenComPort.vi　　ノードのポートを接続します。
- CloseComPort.vi　　ノードのポートを切断します。
- GetData.vi　　接続されたノードから、アナログ・デジタルの値を取得します。
- SetAnalogData.vi　　接続されたノードへアナログデータを出力します。
- SetDigitalData.vi　　接続されたノードへデジタルデータを出力します。
- GetCommandCount.vi　　ノードとの通信回数を取得します。
- GetLastError.vi　　通信時のエラー情報を取得します。
- GetCommData.vi　　使用しません。
- SetCommData.vi　　使用しません。
- SetDebugFlag.vi　　使用しません。

目的別命令

- WagoAnalogInput.vi　　接続されたノードからアナログデータを取得します。
- WagoAnalogOutput.vi　接続されたノードへアナログデータを出力します。
- WagoDigitalInput.vi　接続されたノードからデジタルデータを取得します。
- WagoDigitalOutput.vi　接続されたノードへデジタルデータを出力します。
- WagoSetConfig.vi　　ノードのコンフィグレーションを行います。
- WagoOpenComPort.vi　ノードのポートを接続します。
- WagoClosePort.vi　　ノードのポートを切断します。
- WagoGetCmdCount.vi　ノードとの通信回数を取得します。
- WagoFileWrite.vi　　ログファイルを出力します。

12.1.3　WagoSamples.llb 概要

・WagoPnlConfig.vi　　通信の設定・オープンを行う VI です。カスタマイズして使用できます。
・Wago_AI_Sample.vi　アナログ入力のサンプルプログラムです。
・Wago_AO_Sample.vi　アナログ出力のサンプルプログラムです。
・Wago_DI_Sample.vi　デジタル入力のサンプルプログラムです。
・Wago_DO_Sample.vi　デジタル出力のサンプルプログラムです。

12.2　基本命令－DLL（WAGO_IO.dll）呼び出し用 VI

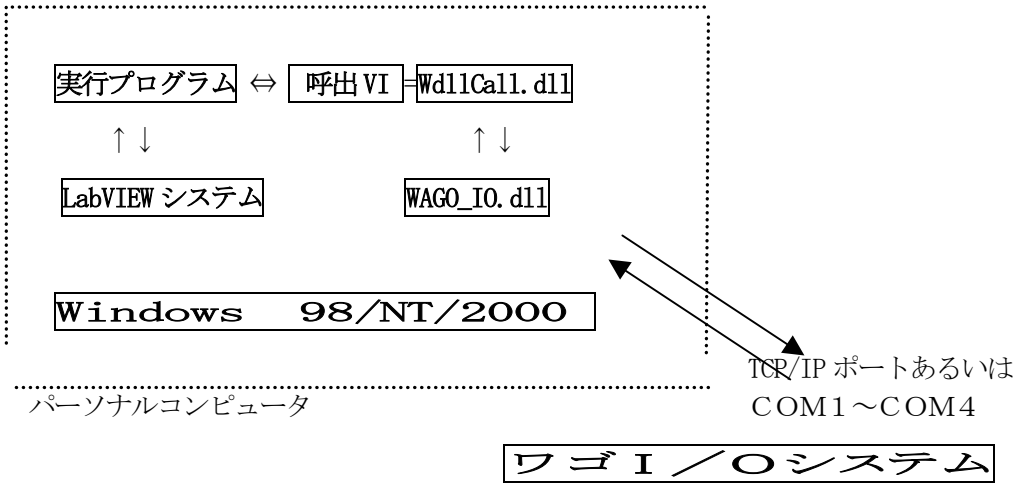
本 DLL 呼び出し用 VI は、標準添付の開発支援 DLL（Dynamic Link Library）を LabVIEW 上から呼び出して使用するためのものです。フォルダーProgramFiles¥Wago_io¥WdllCall¥の下にあるライブラリ WagoIO.llb の中に収納されています。
LabVIEW においてアプリケーションソフトを開発するときに、モジュールの構成（コンフィグレーション）やビットデータの物理量変換、通信プロトコル等を気にすることなく、アプリケーション開発に専念できます。
対応する各 DLL 関数に関しての詳細は WAGO_IO.dll の仕様をご確認ください。

以下の表は WAGO_IO.dll を動作させるための呼び出し用 VI と DLL 関数の対応一覧表です。

	関数名（VI 名／DLL 関数名）	機能・役割
通信動作	OpenComPort.vi ／ WAGO_OpenCommPort	TCP/IP あるいはシリアルポートをオープンするための関数
	CloseComPort.vi ／ WAGO_CloseCommPort	TCP/IP あるいはシリアルポートをクローズのための関数
ﾓｼﾞｭｰﾙ ｺﾝﾌｨｸﾞﾚｰｼｮﾝ	SetConfig.vi ／ WAGO_SetConfig	指定ノードのモジュール構成情報を DLL に通知するための関数
データの 入出力	GetData.vi ／ WAGO_GetData	入力モジュールのデータ読み込み処理関数
	SetDigitalData.vi ／ WAGO_SetDigitalData	ディジタル出力モジュールへのデータ出力関数
	SetAnalogData.vi ／ WAGO_SetAnalogData	アナログ出力モジュールへのデータ出力関数

エラー情報の取得	GetLastError.vi ／ 該当なし	最後に起こったエラー内容を保持します。

以下にパーソナルコンピュータ（PC）内において実行プログラムが作成された場合、このプログラムと共に用いられる各ソフトウェアの構成を示します。



LabVIEW は多種のプラットフォームに対応するために、独自のデータ構造を持ちます。
このシステム上から WAGO_IO.dll を呼び出すために、データ変換用の DLL（Wdllcall.dll、１２－１節）が存在し、これを通して WAGO_IO.dll を呼び出しています。

12.2.1 DLL（WAGO_IO.dll）呼び出し用 VI 関数仕様書

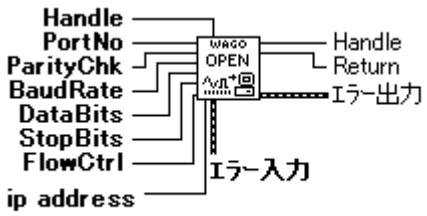
ここではDLL（WAGO_IO.dll）呼び出し用VI各関数の機能およびブロックダイアグラムを説明します。

１．通信に関するVI

（1）関数名： OpenComPort.vi

機能： バスカップラとの通信を行う為にTCP/IPあるいはシリアルポートをオープンします。
通信を行うポートと通信方式を指定して呼び出します。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32 Handle コンフィグを行ったノードのハンドルです。
必ず接続を行ってください。

I16 PortNo オープンするComポートの番号を指定します。

abc ip address Ethernet（TCP/IP）接続するときにIPアドレスを指定します。

I16 ParityChk　シリアル通信時に行うパリティチェックを指定します。
０：無し
１：偶数パリティ　　２：奇数パリティ

I32 BaudRate　通信速度(ボーレート)を指定します。
9600, 19200 等の値を設定します。

I16 DataBits　データビット長を指定します。
８：８ビット長　　７：７ビット長

I16 StopBits ストップビット長を指定します。

I16 FlowCtrl　フローコントロールの指定を行います。
０：無し　　１：Xon/Xoff　　２：ハードウェア

Error もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表¥示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

TF status ｴﾗｰがある場合にはTRUEになります。

I32 code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

abc source ｴﾗｰがどのVIで起こったかを示します。

I32 Return　Wago_IO.dllの WAGO_OpenCommPort() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

I32 Handle コンフィグを行い、オープンされたハンドルです。
このノードに対してアクセスするには必ず接続してください。

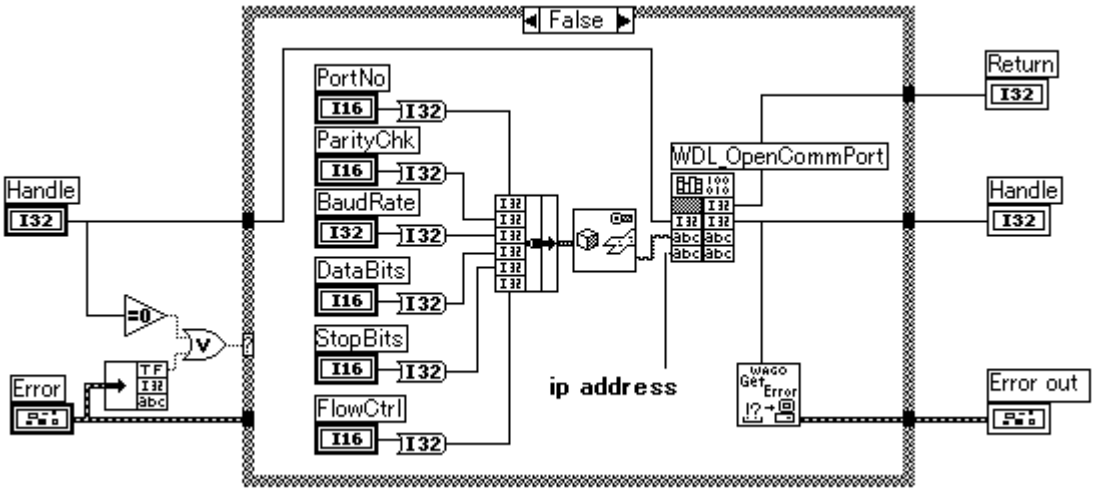
Error out　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。
それ以外はこのVI内で起こったｴﾗｰを示します。

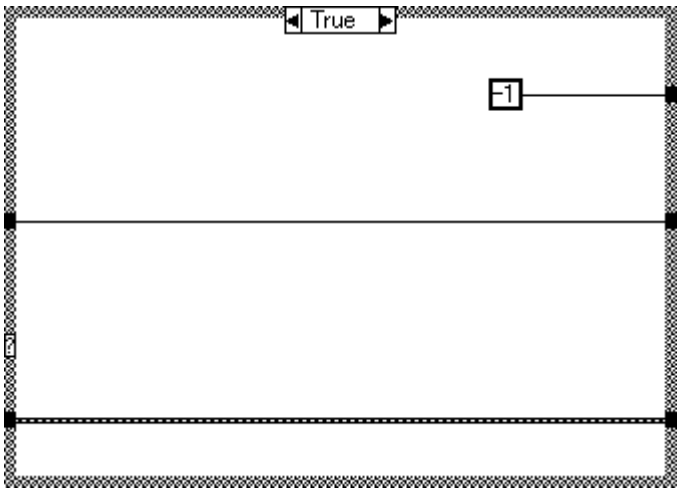
TF status ｴﾗｰがある場合にはTRUEになります。

I32 code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

abc source ｴﾗｰがどのVIで起こったかを示します。

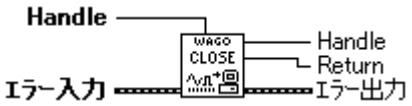
ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ





（2）関数名： CloseComPort.vi
機能： シリアルポートをクローズします。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32 Handle コンフィグ後にオープンを行ったノードのハンドルです。
ここに指定されたハンドルをクローズします。

Error もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表¥示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

TF status ｴﾗｰがある場合にはTRUEになります。

I32 code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

abc source ｴﾗｰがどのVIで起こったかを示します。

I32 Return Wago_IO.dllの WAGO_CloseCommPort() の戻り値です。
０の場合は正常終了です。
負の値の場合はエラーを表します。

I32 Handle コンフィグを行い、オープンされたハンドルです。
Close命令以降は０になります。

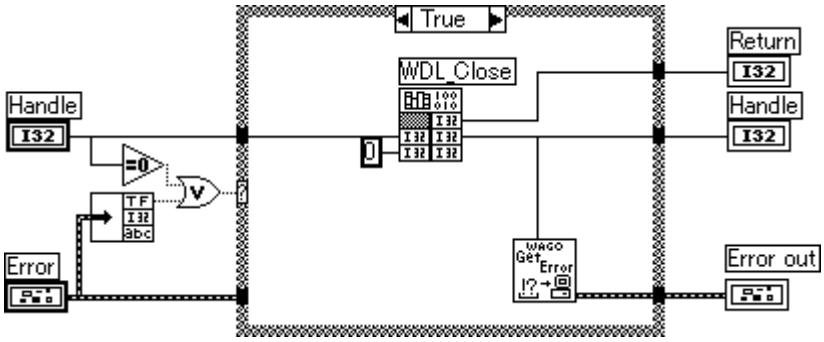
Error out　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表¥示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

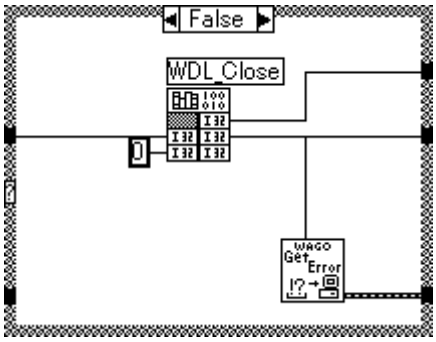
TF status ｴﾗｰがある場合にはTRUEになります。

I32 code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

abc source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ

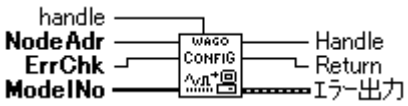




２．モジュール構成（コンフィグレーション情報）通知に関するVI

関数名： SetConfig.vi
機能： 指定ノードのコンフィグレーション情報を設定します。

ｺﾈｸﾀﾍﾟｰｼﾞ



制御器と表示器:

I32 handle　ハンドルを保持する領域を確保します。

I16 NodeAdr　ノードアドレスの指定です。設定するノードの番号は１からです。
TCP/IPで複数のノードを接続するときはポート識別番号として使います。

I16 ErrChk　ＣＲＣ／ＬＲＣチェック　有無フラグ
０の時はエラーチェックを行いません。
エラーチェックを行う時は０以外の値を設定してください。

[U32] ModelNo　モデル番号の配列です。
使用するモジュール番号をバスカプラか順番に設定してください。

U32 **モジュール型番** 使用するモジュールの型番号

I32 Return　Wago_IO.dllの WAGO_SetConfig() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

I32 Handle　ノードに付けられたハンドルです。
以降のWagoViに対しては必ず引き渡してください。

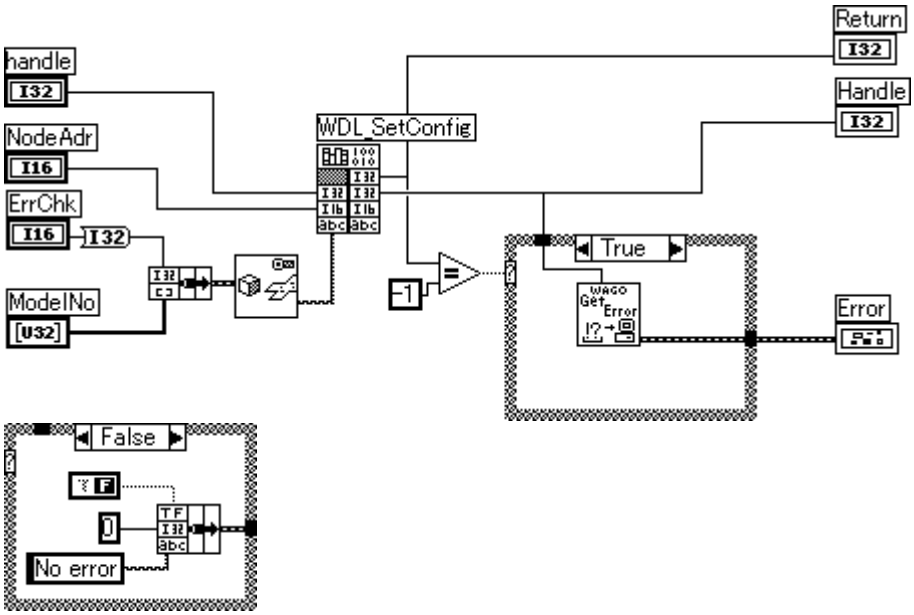
Error out　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。
それ以外はこのVI内で起こったｴﾗｰを示します。

TF status ｴﾗｰがある場合にはTRUEになります。

I32 code ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

abc source ｴﾗｰがどのVIで起こったかを示します。

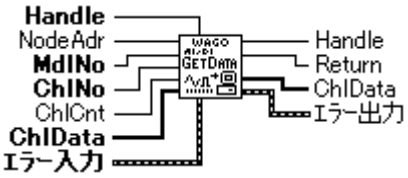
ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



３．データ入力（取得）に関するVI

関数名：GetData.vi
機能：　入力モジュールのデータ読み込み処理関数
　　アナログ・デジタルの両方のモジュールを同時に扱います。
対象：　750-400、750-401、750-402、750-403、750-405、750-406、750-408、750-409、750-410
　　750-412、750-414、750-415、750-452、750-454、750-456、750-461、750-462、750-465
　　750-466、750-467、750-468、750-469、750-472、750-474、750-479、750-480、750-476、
　　750-478　および750-506の診断データ

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32　Handle　コンフィグを行ったノードのハンドルです。
必ず接続を行ってください。

I16　NodeAdr　操作を行うノード番号を指定します。TCP/IPで複数ノードを想定する場合はポート識別番号として使用します。

I16　MdlNo　指定範囲を決める時に、その数え始めのモジュール位置を整数で指定します（バスカップラに近い方が１）。
７５０－４０４を奇数個使用するときには、７５０－４０４をバスカプラから一番遠い方に設置して、７５０－６２２をディジタル出力モジュールの先頭の左側（バスカップラに近い方）に挿入してください。このとき、７５０－６２２はモジュールとしては数えません。

I16　ChlNo　MdlNoで指定されたモジュール内で、何番目のチャンネルから数え始めるかを整数で指定します（基数は１）。

I16　ChlCnt　ChlNoで指定したチャンネルから同時に入力を行う入力チャンネル数（範囲）を整数で設定します。
アナログ･デジタル両方のモジュールが対象となります。

[SGL]　ChlData　デジタル・アナログモジュールから取得するためのデータ配列です。
ここに設定した数だけのデータ領域が確保され、データを取得できます。

　SGL　**確保データ領域** モジュールから取り込むためのデータ領域は、ここでデータを設定しておかないと更新されません（値は使用しません）。

Error　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

　TF　Status　ｴﾗｰがある場合にはTRUEになります。

　I32　code　ｴﾗｰの種類を示す数値です。0はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

　abc　source　ｴﾗｰがどのVIで起こったかを示します。

I32 **Return**　Wago_IO.dllの WAGO_GetData() の戻り値です。
0の場合は正常動作を表し、負の値の場合はエラーを表します。

I32 **Handle** コンフィグを行い、オープンされたハンドルです。
このノードに対してアクセスするには必ず接続してください。

DBL **ChlData**　デジタル・アナログモジュールから取得されたデータ配列です。
入力設定数のデータ領域のみ、データを取得できます。

　DBL **取得データ** モジュールから取得した各チャンネルのデータです。

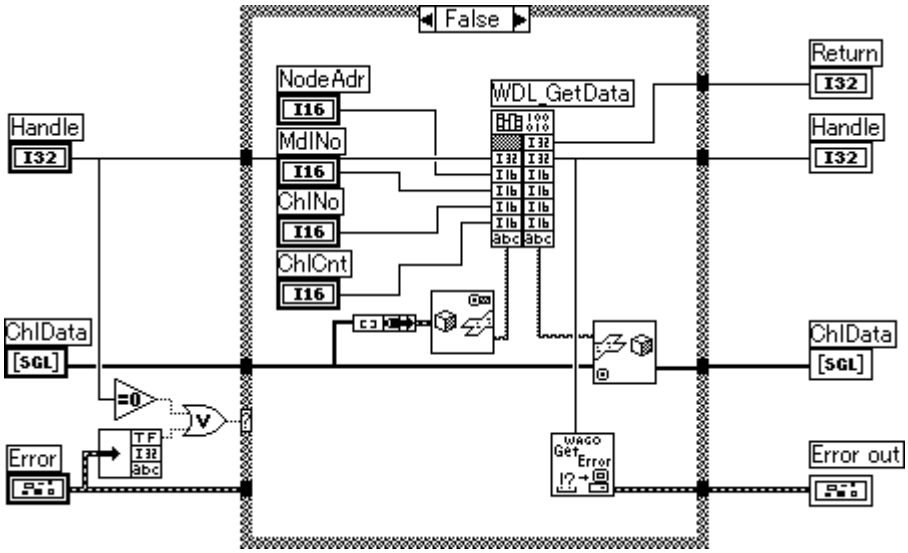
Error out　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。
それ以外はこのVI内で起こったｴﾗｰを示します。

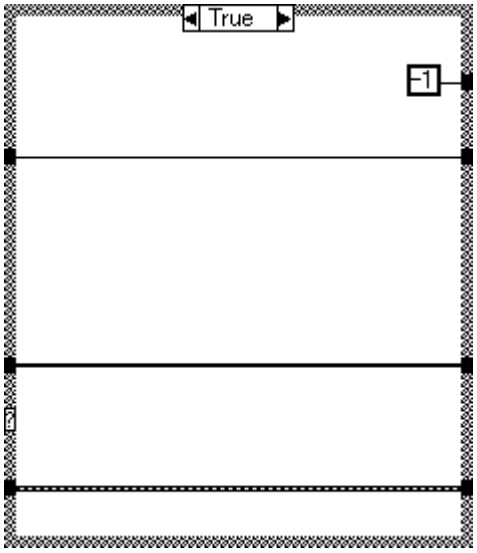
　TF **status** ｴﾗｰがある場合にはTRUEになります。

　I32 **code** ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

　abc **source** ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ





1）取得されたデータはデジタル・アナログモジュールに関係無くチャンネル毎に独立した形で得られます。ディジタルモジュールの各チャンネルのデータはON が「１」、OFF が「０」で表わされます。

2）アナログデータは物理単位そのままの値を表わします。単位については　「V」,「mA」,「℃」がありますが、それぞれのモジュールに対応した物理単位で処理してください。

3）７５０－５０６の診断データについては各チャンネルごとに以下のようなビット構成になっていますが、データは各々のチャンネルごとに０～３の 10 進数で表わすことができます。

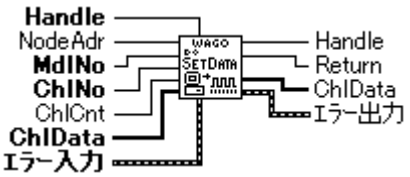
	Bit 3	Bit 2	Bit 1	Bit 0
機　　能	Ch2 出力の診断		Ch1 出力の診断	
正常	0	0	0	0
負荷が接続されていない（または断線）	0	1	0	1
短絡（出力回路のショート）	1	0	1	0
電源電圧が極端に低い	1	1	1	1

４．デジタル出力に関するVI

関数名：SetDigitalData.vi
機能：　デジタル出力モジュールへのデータ出力処理関数です。
　　複数のデジタル出力モジュールを同時に扱うことが可能です。

対象：　750-501、750-502、750-504、750-506、750-509、750-516、750-512、750-513、750-514

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32　Handle　コンフィグを行ったノードのハンドルです。
必ず接続を行ってください。

I16　NodeAdr 操作を行うノード番号を指定します。TCP/IPで複数のノードを制御するときはポート識別番号として使用します。

I16　MdlNo 指定範囲を決める時に、その数え始めのモジュール位置を整数で指定します（バスカップラに近い方が１）。

I16　ChlNo MdlNoで指定されたモジュール内で、何番目のチャンネルから数え始めるかを整数で指定します（基数は１）。

I16　ChlCnt　ChlNoで指定したチャンネルから同時に出力を行う出力チャンネル数（範囲）を整数で設定します。
但しデジタルモジュールだけが対象となります。

[SGL]　ChlData　デジタルモジュールへ出力するデータ配列です。

　SGL　出力データ デジタルモジュールへ出力するデータです。

Error もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

　TF　Status ｴﾗｰがある場合にはTRUEになります。

　I32　code ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

　abc　source ｴﾗｰがどのVIで起こったかを示します。

I32　Return　Wago_IO.dllの WAGO_SetDigitalData() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

I32　Handle コンフィグを行い、オープンされたハンドルです。

このノードに対してアクセスするには必ず接続してください。

[SGL] ChlData　デジタルモジュールへ出力するデータ配列です。
出力されたデータがそのまま残ります。
接続の必要はありません。

[SGL] 出力データ デジタルモジュールへ出力するデータのコピーです。

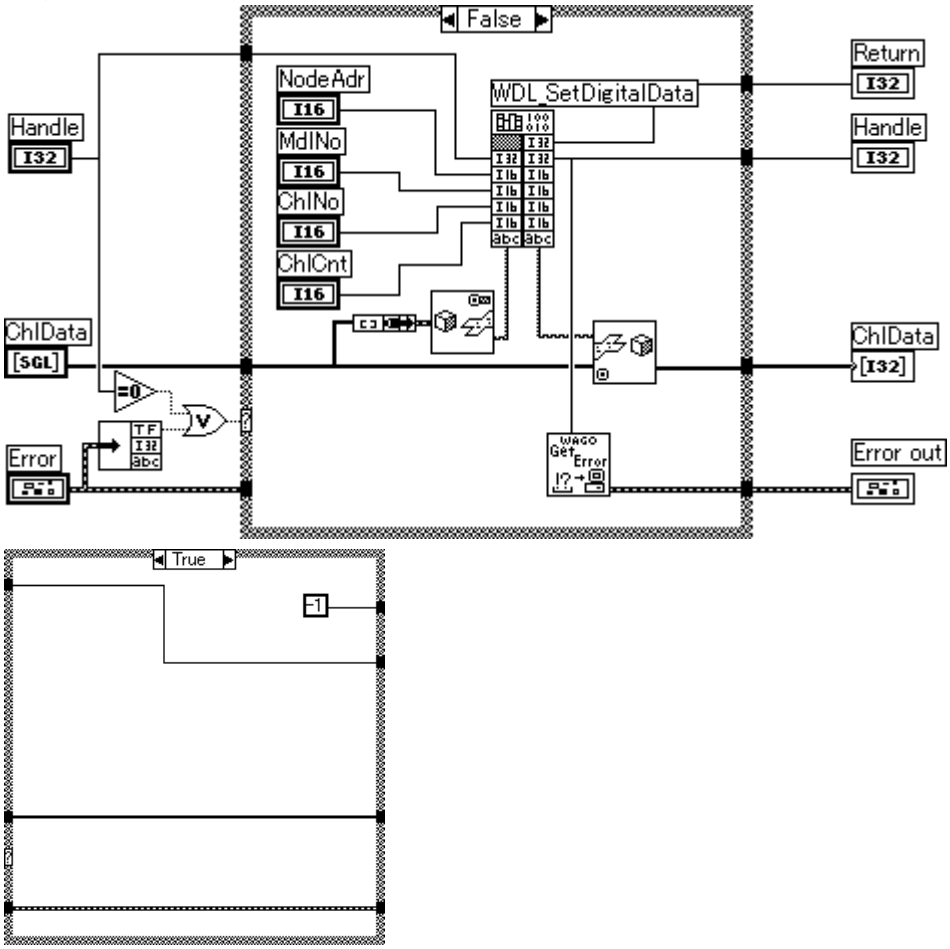
[ERR] Error out　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。
それ以外はこのVI内で起こったｴﾗｰを示します。

[TF] status ｴﾗｰがある場合にはTRUEになります。

[I32] code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

[abc] source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ

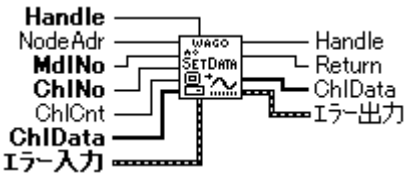


５．アナログ出力に関するVI

関数名：SetAnalogData.vi
機能：　アナログ出力モジュールへのデータ出力処理を行います。
　　複数のアナログ出力モジュールを同時に処理することができます。

対象：　750-550、750-552、750-554、750-556

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32　Handle　コンフィグを行ったノードのハンドルです。
必ず接続を行ってください。

I16　NodeAdr 操作を行うノード番号を指定します。TCP/IP接続で複数のノードを制御するときはポート識別番号として使用します。

I16　MdlNo 指定範囲を決める時に、その数え始めのモジュール位置を整数･ｲ指定します（バスカップラに近い方が１）。

　I16　ChlNo MdlNoで指定されたモジュール内で、何番目のチャンネルから数え始めるかを整数で指定します（基数は１）。

I16　ChlCnt　ChlNoで指定したチャンネルから同時に出力を行う出力チャンネル数（範囲）を整数で設定します。
但しアナログモジュールだけが対象となります。

[SGL]　ChlData アナログモジュールへ出力するデータ配列です。

SGL　出力データ アナログモジュールへ出力するデータです。

Error もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

　TF　status ｴﾗｰがある場合にはTRUEになります。

　I32　code ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

　abc　source ｴﾗｰがどのVIで起こったかを示します。

I32　Return Wago_IO.dllの WAGO_SetAnalogData() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

I32　Handle コンフィグを行い、オープンされたハンドルです。

このノードに対してアクセスするには必ず接続してください。

[SGL] ChlData アナログモジュールへ出力するデータ配列です。
出力されたデータがそのまま残ります。接続の必要はありません。

[SGL] 出力データ アナログモジュールへ出力するデータのコピーです。

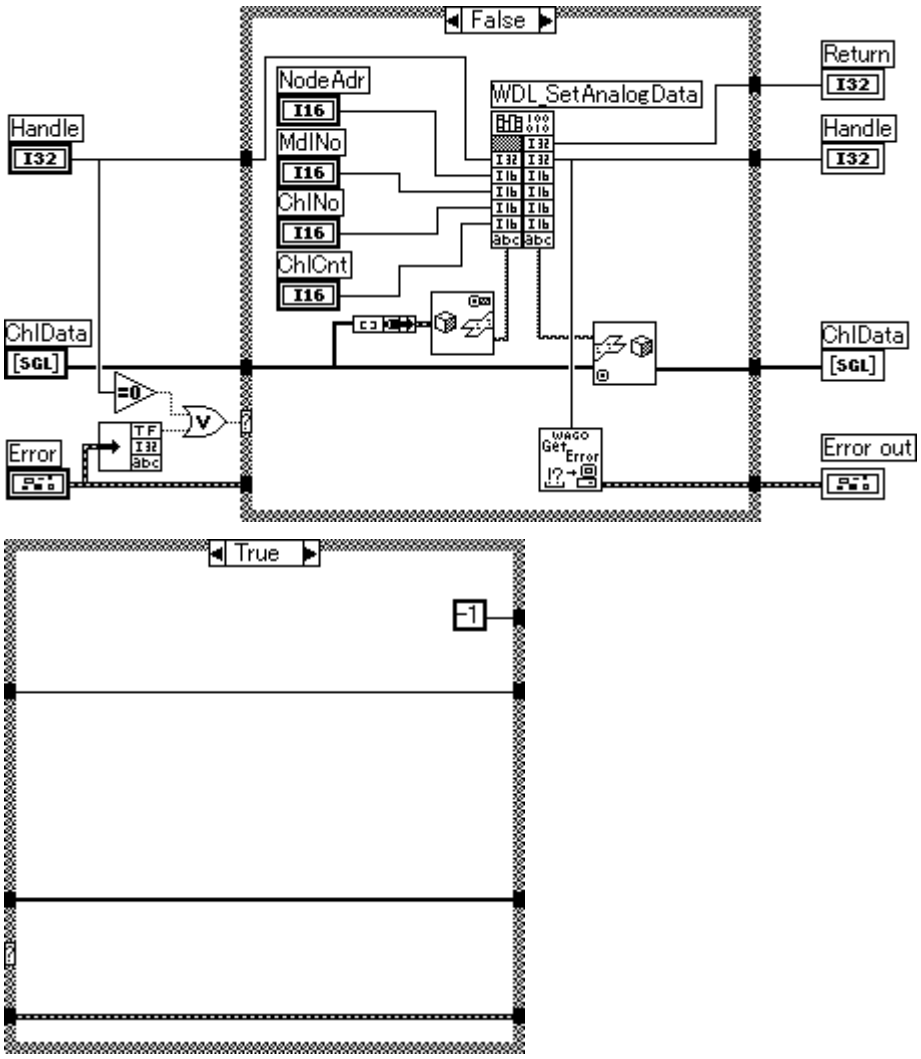
Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。
それ以外はこのVI内で起こったｴﾗｰを示します。

[TF] status ｴﾗｰがある場合にはTRUEになります。

[I32] code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

[abc] source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



1） それぞれのモジュールの設定範囲は以下のとおりです。プログラム記述上では数値分解能についての制約はありませんが、アナログモジュールの分解能は１２ビットですので実際に有効な最小単位は以下のとおりです。

モデル名	出力範囲	有効な最小単位
７５０－５５０	０～10　V	10／4096 ＝　0.0024　V
７５０－５５２	０～20　mA	20／4096 ＝　0.0048　mA
７５０－５５４	４～20　mA	16／4096 ＝　0.0039　mA
７５０－５５６	－10～+10　V	10／4096 ＝　0.0024　V

６．コマンド回数確認のためのVI

関数名：GetCommandCount.vi
機能：　コマンド回数確認
指定されたノードのバスカップラから実行されたコマンドの回数を読み出します。
（正常に行われたコマンド回数がバスカップラのメモリカウンタにカウントされます。
バスカップラの電源をオフにするとこのメモリカウンタはリセットされます。）

ｺﾈｸﾀﾍﾟｰﾝ

制御器と表示器

I32 **Handle**　コンフィグを行ったノードのハンドルです。
必ず接続を行ってください。

I16 **NodeAdr** 操作を行うノード番号を指定します。TCP/IPで複数のノードを想定するときはポート識別番号として使用します。

Error もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

- TF **status** ｴﾗｰがある場合にはTRUEになります。
- I32 **code** ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc **source** ｴﾗｰがどのVIで起こったかを示します。

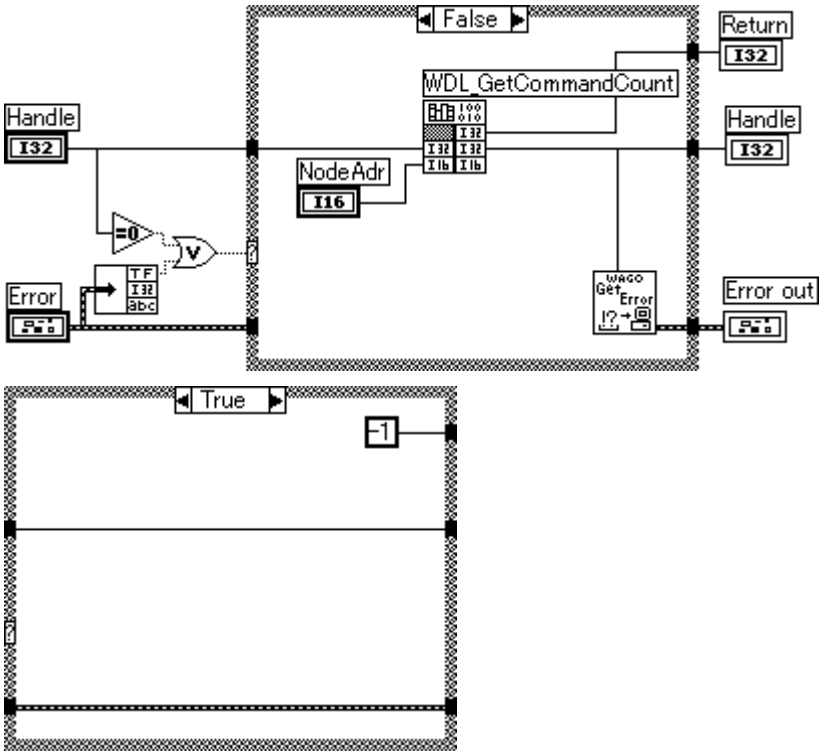
I32 **Return**　Wago_IO.dllの WAGO_GetCommandCount() の戻り値です。
正の値または０の場合は受け取ったコマンドの数を表します。
負の値の場合はエラーを表します。

I32 **Handle** コンフィグを行い、オープンされたハンドルです。
このノードに対してアクセスするには必ず接続してください。

Error out　もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。
それ以外はこのVI内で起こったｴﾗｰを示します。

- TF **status** ｴﾗｰがある場合にはTRUEになります。
- I32 **code** ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc **source** ｴﾗｰがどのVIで起こったかを示します。

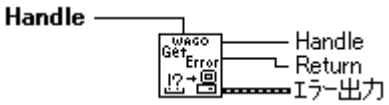
ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



８．エラー処理のためのVI

関数名：GetLastError.vi
機能：　最終エラー情報の取得
　　指定したハンドルを使用して発生したエラーの情報を取得します。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32 Handle　コンフィグを行ったノードのハンドルです。
必ず接続を行ってください。

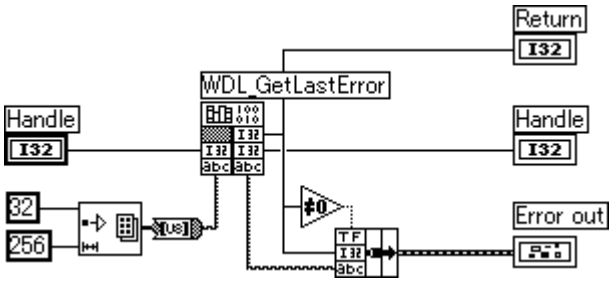
I32 Return　WdllCall.dllの WDL_GetLastError() の戻り値です。
この関数は各ハンドルで発生したエラーNoを保持しています。
エラーがない場合は0を返します。

I32 Handle コンフィグを行い、オープンされたハンドルです。
このノードに対してアクセスするには必ず接続してください。

Error out 各ハンドルにおいて、最後に起こったエラーコードを保持します。

- TF status ｴﾗｰがある場合にはTRUEになります。
- I32 code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、数字はｴﾗｰｺｰﾄﾞを示します。
- abc source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



返されるエラーコードとメッセージ
各ハンドルにおいて、最初にエラーが発生した関数を記憶しています。

WDLERRMSG_NOERROR	0	"no error"
WDLERR_CONFIG	99001	"Config error"
WDLERR_OPEN	99002	"Open error"
WDLERR_CLOSE	99003	"Close error"
WDLERR_GETCOMCOUNT	99004	"GetComCount error"
WDLERR_GETDATA	99010	"GetData error"
WDLERR_SETANDATA	99011	"SetAnalogData error"
WDLERR_SETDIDATA	99012	"SetDigitalData error"
WDLERRMSG_OTHERERROR	その他	"Invalid operation error"

12.3 目的別命令－WagoIO 入出力 VI

本 WagoIO 入出力 VI は、ワゴ I/O のデータ入出力に関しユーザープログラムの作成をより簡単にするために用意されたものです。フォルダーProgramFiles¥Wago_io¥WdllCall¥の下にある WagoIO.llb の中に収納されています。

WAGO_IO.dll を呼び出す基本命令の組み合わせでは簡単なアプリケーションを作成、テストする為に煩雑な処理や作成に時間がかかることがあります。そこで Wago_io 用 VI 基本命令をより簡単に使用するためパラメータをまとめた「入出力クラスタ」を作成し、これを **WagoIO 入出力 VI** に入れました。この VI を用いますと I/O の取り扱いがより容易になります。

次節では以下の関数についての情報が記載されています。

関数名：WagoAnalogInput.vi	アナログの入力を取り込みます。
関数名：WagoAnalogOutput.vi	アナログの出力を行います。
関数名：WagoDigitalInput.vi	デジタルの入力を取り込みます。
関数名：WagoDigitalOutput.vi	デジタルの出力を行います。
関数名：WagoFileWrite.vi	取り込んだデータをログファイルとして出力します。
関数名：WagoSetConfig.vi	コンフィグ設定を行い、入出力クラスタを作成します。
関数名：WagoOpenComPort.vi	入出力クラスタを元にポートをオープンします。
関数名：WagoClosePort.vi	入出力クラスタを元にポートをクローズします。
関数名：WagoGetCmdCount.vi	入出力クラスタを元に通信回数を確認します。

入出力クラスタ： 作成されたノードのハンドル、ノードアドレス、エラー出力クラスタなどをまとめた物です。各パラメータを取り出すには Unbundle, Unbundle by name を使用してください。

入出力クラスタの構成

入出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決
- I32 **Handle** ノードを設定した時に作成され、ユニークなコードを持ちます。
- I16 **NodeAdr** 操作の対象となるノード番号、あるいはポート識別番号
- I16 **MdlNo** 操作の対象となるモジュール番号
- I16 **CnlNo** 操作の対象となるチャンネル番号
- **ｴﾗｰ出力** もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されま

す。それ以外はこのVI内で起こったｴﾗｰを示します。

TF **ｽﾃｰﾀｽ** ｴﾗｰがある場合にはTRUEになります。

I32 **ｺｰﾄﾞ** ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

abc **ｿｰｽ** ｴﾗｰがどのVIで起こったかを示します。

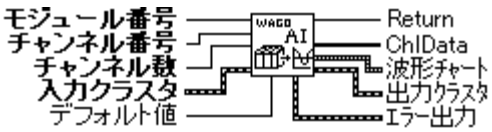
12.3.1 WagoIO 入出力 VI 関数仕様書

１．アナログ入力に関するVI

関数名：WagoAnalogInput.vi
機能：アナログ入力モジュールのデータ読み込み処理関数

アナログデータは物理単位そのままの値を表します。
単位は「V」「mA」「℃」などそれぞれのモジュールに対応した物理単位で処理してください。
取り込まれたデータは、直接波形チャートに接続できます。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I16 **モジュール番号** モジュールの番号を指定します。
I16 **チャンネル番号** モジュール内のチャンネルを指定します。
I32 **チャンネル数** 操作を行うチャンネルの数を指定します。
SGL **デフォルト値** 確保される領域（チャンネル数で指定）に設定する初期値
入力ｸﾗｽﾀ ハンドルやノードアドレス、及びエラー情報を保持します。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

I32 **Return** Wago_IO.dllの WAGO_GetData() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。
[SGL] **ChlData** 各アナログチャネルの取得データを表示します。
指定チャンネルの入力で、初期値・エラー時は０です。
波形ﾁｬｰﾄ LabVIEWの標準部品VIを使用しています。
ユーザーのカスタマイズした波形チャートなどに接続してください。
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。処理を行った結果が格納され、次の関数に引き渡されます。
- I32 Handle
- I16 NodeAdr

MdlNo

CnlNo

ｴﾗｰ出力

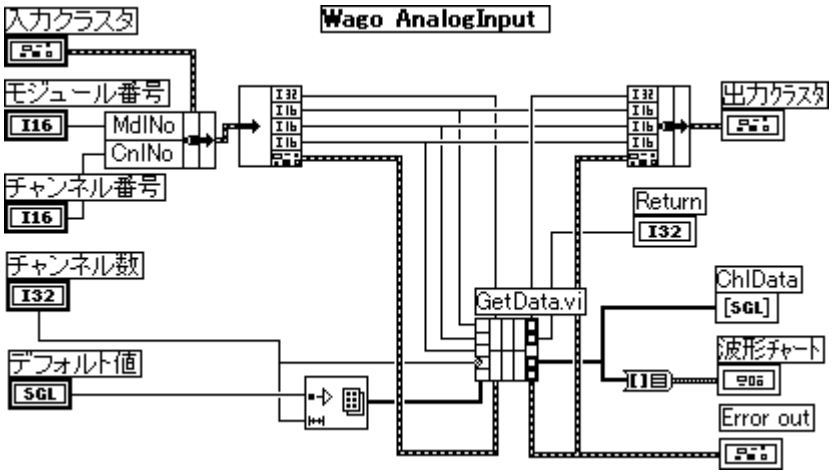
Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

status ｴﾗｰがある場合にはTRUEになります。

code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。

source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



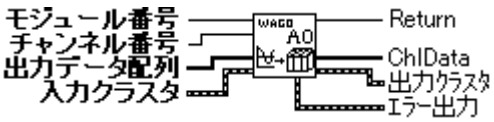
２．アナログ出力に関するVI

関数名：WagoAnalogOutput.vi

機能：アナログモジュールへのデータ出力

複数のアナログ出力モジュールを同時に処理することができます。
・それぞれのモジュールの設定範囲はプログラム上では制限はありませんが、アナログモジュールの分解能は１２ビットになっております。
実際に有効な最小単位は以下に示す値になります。
モデル７５０－５５０ 出力範囲　０～１０V　有効な最小単位　１０/4096 ＝ 0.0024V
モデル７５０－５５２ 出力範囲　０～２０mA　有効な最小単位　２０/4096 ＝ 0.0048mA
モデル７５０－５５４ 出力範囲　４～２０mA　有効な最小単位　１６/4096 ＝ 0.0039mA
モデル７５０－５５６ 出力範囲 －１０～１０V　有効な最小単位　１０/4096 ＝ 0.0024V

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

- I16 **モジュール番号** モジュールの番号を指定します。
- I16 **チャンネル番号** モジュール内のチャンネルを指定します。
- SGL **出力データ配列** 各アナログチャネルの出力データ
- **入力ｸﾗｽﾀ** ハンドルやノードアドレス、及びエラー情報を保持します。
  - I32 Handle
  - I16 NodeAdr
  - I16 MdlNo
  - I16 CnlNo
  - **ｴﾗｰ出力**

I32 **Return** Wago_IO.dllの WAGO_SetAnalogData() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

SGL **ChlData** 各アナログチャネルの出力データ

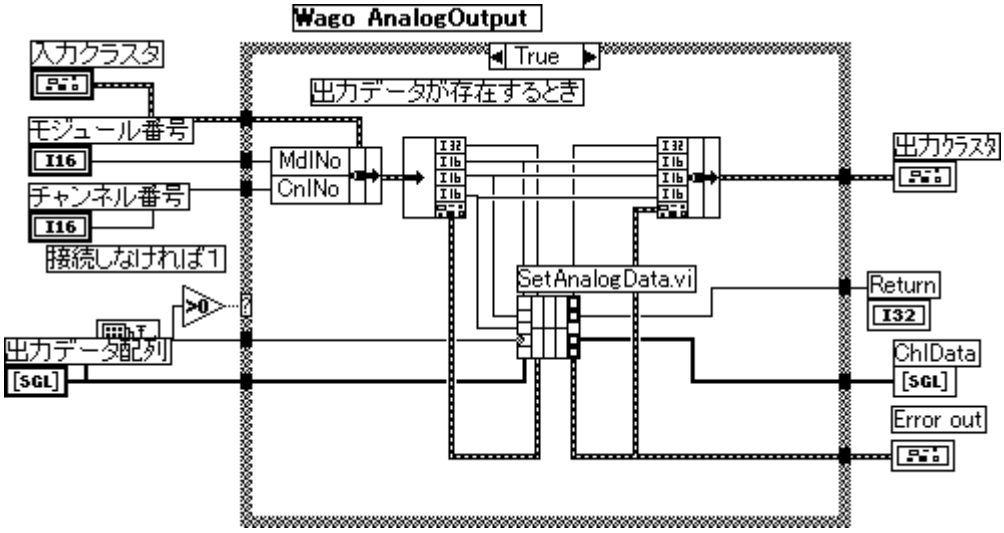
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
処理を行った結果が格納され、次の関数に引き渡されます。

- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

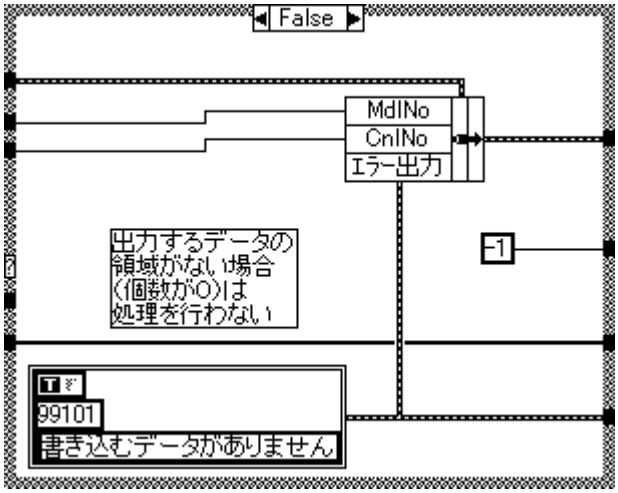
Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

- TF **status** ｴﾗｰがある場合にはTRUEになります。
- I32 **code** ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc **source** ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



配列の要素がない場合、処理を行いません。



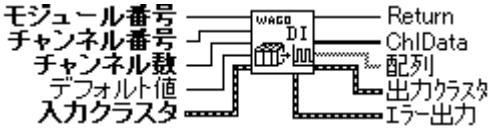
３．デジタル入力に関するVI

関数名：WagoDigitalInput.vi

機能：デジタル入力モジュールのデータ読み込み処理関数

・取得したデータはチャンネル毎に独立した形で得られます。
数値（Onが１、Offが０）とBOOL値（OnがTRUE,OffがFALSE）の出力があります。
ブール値（ランプ等）の配列に直接接続できます。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

- I16 **モジュール番号** モジュールの番号を指定します。
- I16 **チャンネル番号** モジュール内のチャンネルを指定します。
- I32 **チャンネル数** 操作を行うチャンネルの数を指定します。
- I32 **デフォルト値** 確保される領域（チャンネル数で指定）に設定する初期値
- **入力ｸﾗｽﾀ** ハンドルやノードアドレス、及びエラー情報を保持します。
  - I32 Handle
  - I16 NodeAdr
  - I16 MdlNo
  - I16 CnlNo
  - **ｴﾗｰ出力**

I32 **Return** Wago_IO.dllの WAGO_GetData() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

[SGL] **ChlData** 各デジタルチャンネルの取得データを表示します。
１.０はOnを表し、初期値・エラー時は０です。

[TF] **配列** デジタルの入力値。ブール配列として取り込まれます。

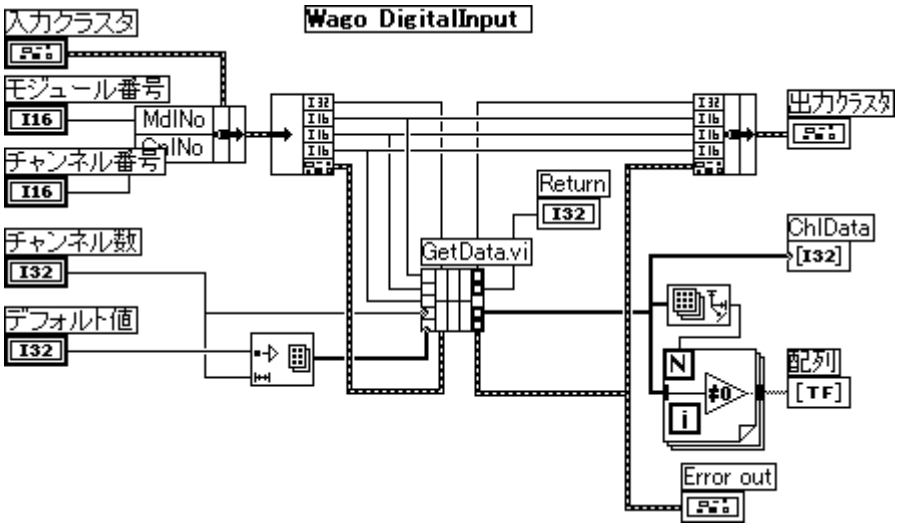
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
処理を行った結果が格納され、次の関数に引き渡されます。

- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

- TF **status** ｴﾗｰがある場合にはTRUEになります。
- I32 **code** ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc **source** ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ


Wago DigitalInput
入力クラスタ
モジュール番号
I16
MdlNo
ChlNo
チャンネル番号
I16
チャンネル数
I32
デフォルト値
I32
GetData.vi
Return
I32
出力クラスタ
ChlData
[I32]
配列
[TF]
Error out

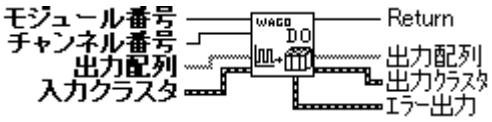
４．デジタル出力に関するVI

関数名：WagoDigitalOutput.vi

機能：デジタル出力モジュールへのデータ出力処理

複数のデジタルモジュールを同時に扱うことが可能です。
出力するデータはブール値（スイッチ等）の配列で引き渡します。

ｺﾈｸﾀﾍﾟｰﾝ



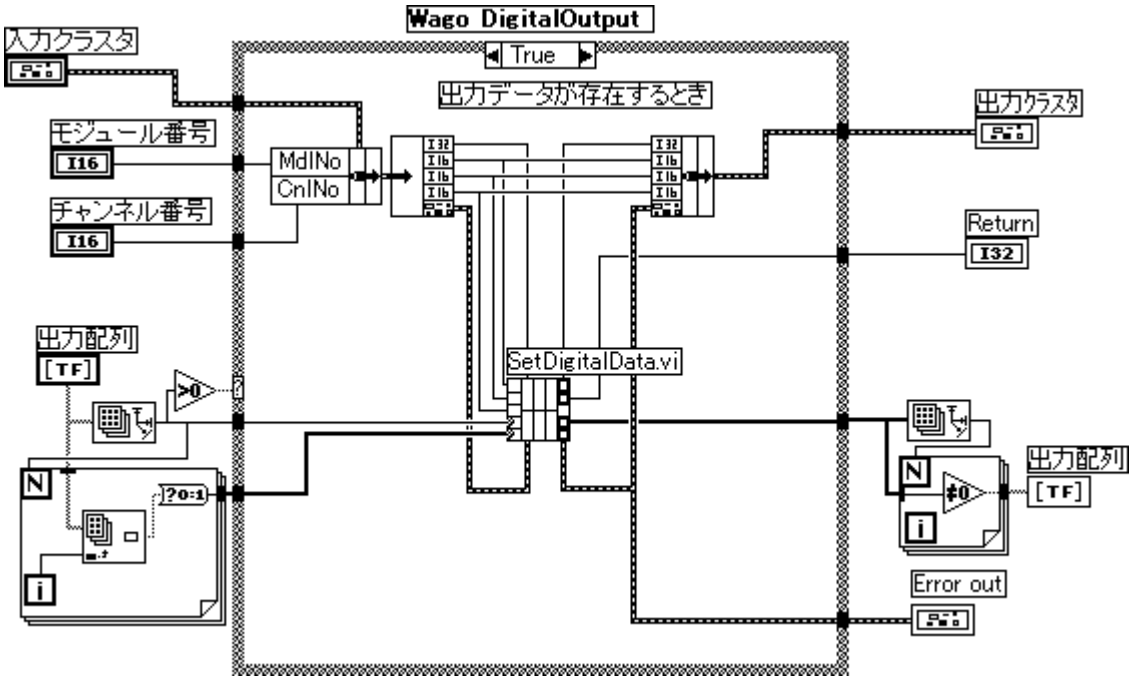
制御器と表示器

I16 **モジュール番号** モジュールの番号を指定します。
I16 **チャンネル番号** モジュール内のチャンネルを指定します。
[TF] **出力配列** 出力するデジタルデータ。ブール配列で指定します。
入力ｸﾗｽﾀ ハンドルやノードアドレス、及びエラー情報を保持します。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- ｴﾗｰ出力

I32 **Return** Wago_IO.dllの WAGO_SetDigitalData() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。
[TF] **出力配列** 出力されたデジタルデータ（ブール配列）
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
処理を行った結果が格納され、次の関数に引き渡されます。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- ｴﾗｰ出力

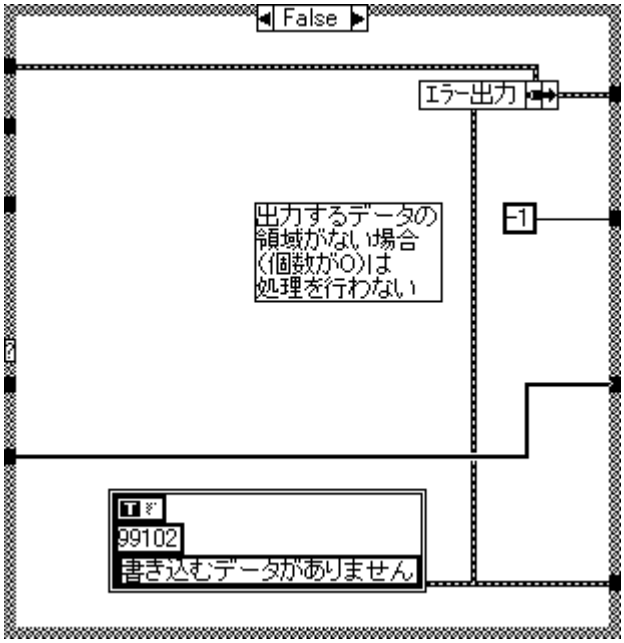
Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。
- TF **status** ｴﾗｰがある場合にはTRUEになります。
- I32 **code** ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc **source** ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



出力するチャンネルの個数は、配列の要素数から算出しています。

配列要素がない場合



５．ログファイル出力に関するVI

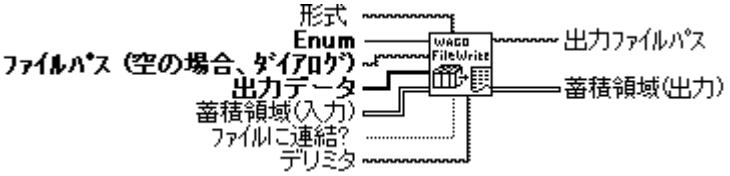
関数名：WagoFileWrite.vi

機能：引き渡されたデータをもとにログファイルを出力

コマンドを指定することで、処理を指定できます。
・none 何もしない
・write ログファイル出力（毎回出力、一回分）
・pool データの蓄積（出力は無し）
・writeall 蓄積データのログファイル出力

データの形式はcsv形式（カンマで区切られたテキスト）で保存します。
※　数値データのみの出力です。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

Enum ログファイル作成時のコマンドを指定します。
・none 何もしない
・write 毎回出力
・pool データの蓄積
・writeall 蓄積データの出力
指定する際は「定数を作成」すると選択できます。

file path ﾌｧｲﾙﾊﾟｽはﾌｧｲﾙのﾊﾟｽ名です。ﾌｧｲﾙﾊﾟｽが空の場合（ﾃﾞﾌｫﾙﾄ設定）またはﾊﾟｽが正しくない場合、VIはﾌｧｲﾙを選択出来る様ﾌｧｲﾙﾀﾞｲｱﾛｸﾞﾎﾞｯｸｽを開きます。ﾕｰｻﾞｰがﾀﾞｲｱﾛｸﾞをｷｬﾝｾﾙした場合、ｴﾗｰ43 が発生します。

出力データ 取り込んだデータ
コマンドによって、ログ出力、データの蓄積が行われます。

蓄積領域(入力)　データを蓄積する為の領域
pool の場合はこの最後に今回のデータが追加されます。

format (%.3f) 形式は数値をどのようにｷｬﾗｸﾀｰに変換するかを指定します。形式文字列が%3f（ﾃﾞﾌｫﾙﾄ）の場合、VIは小数点以下第３位まで含まれる数値の文字列を作成します。形式が%dの場合、VIは全ての数値を含むのに必要なｷｬﾗｸﾀｰ数を使用し、ﾃﾞｰﾀを整数形式に変換します。詳しくは関数とVIﾘﾌｧﾚﾝｽﾏﾆｭｱﾙの第６章を参照して下さい。

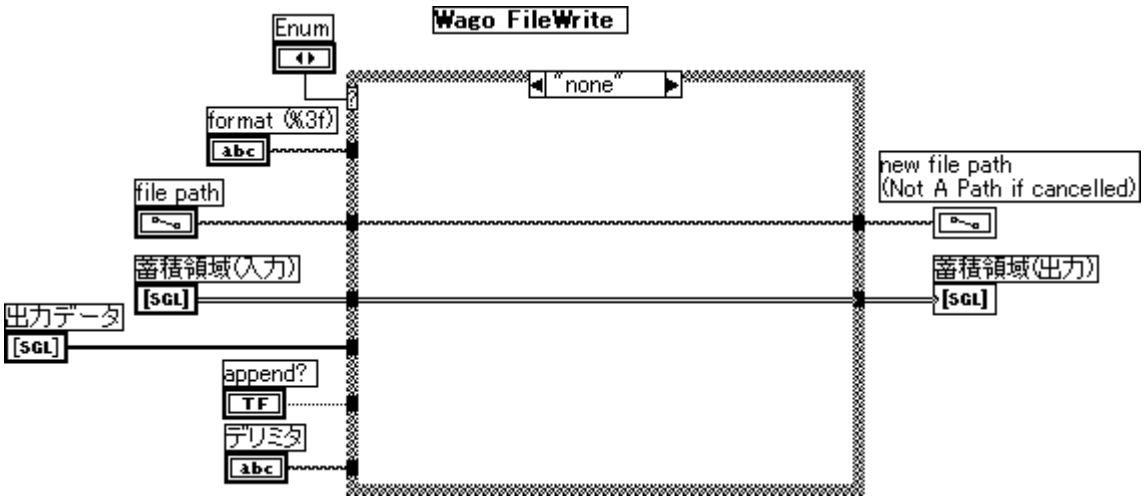
append?「ﾌｧｲﾙに連結」は、ﾃﾞｰﾀを既存ﾌｧｲﾙに追加する場合はTRUEにします。TRUEにして新規ﾌｧｲﾙに書き込むこともできます。ﾃﾞｰﾀを新規ﾌｧｲﾙに書き込むか、既存ﾌｧｲﾙを入れ換える場合は、FALSE(ﾃﾞﾌｫﾙﾄ値)にします。

デリミタ 各数値を区切る記号。通常はカンマを使用してください。

new file path　(Not A Path if cancelled)　「新規ﾌｧｲﾙﾊﾟｽ」は、VIがﾃﾞｰﾀを書き込んだﾌｧｲﾙのﾊﾟｽです。この出力を使用して、ﾀﾞｲｱﾛｸﾞを使用して開くﾌｧｲﾙのﾊﾟｽを決定することができます。ﾕｰｻﾞがﾀﾞｲｱﾛｸﾞﾎﾞｯｸｽでｷｬﾝｾﾙを選択した場合、「新規ﾌｧｲﾙﾊﾟｽ」はNot A Pathになります。

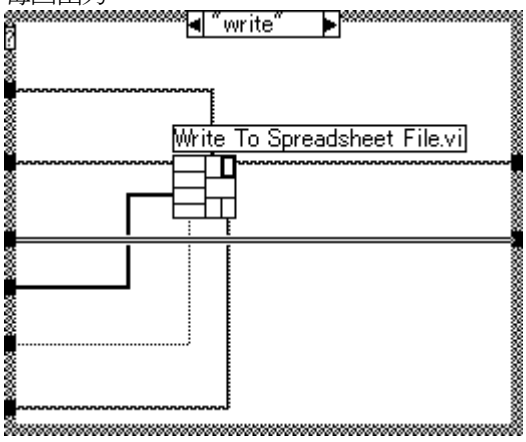
蓄積領域(出力)　蓄積されたデータ
シフトレジスタなどに接続して次回の入力に使います。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ

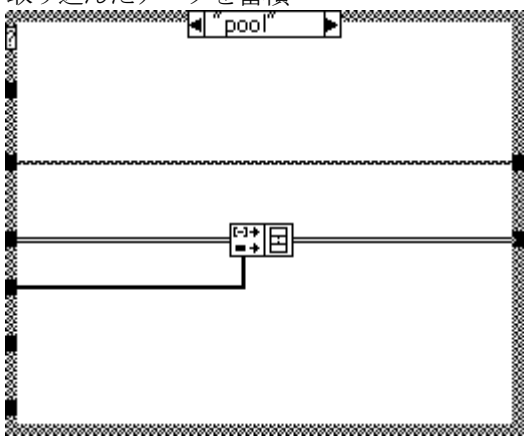


コマンドによって処理が分岐されます。

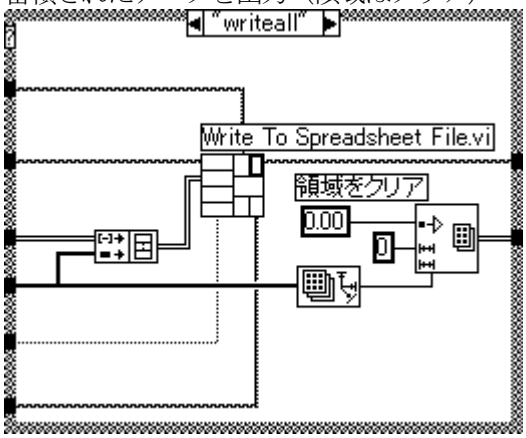
毎回出力



取り込んだデータを蓄積



蓄積されたデータを出力（領域はクリア）



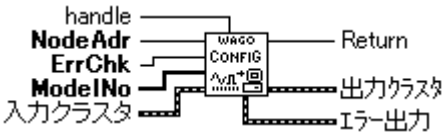
６．ポートの設定に関するVI（クラスタ対応版）

関数名：WagoSetConfig.vi

機能：ノードのコンフィグレーションを設定

指定ノードのコンフィグレーション情報を設定し、各WagoVIに接続する入出力クラスタを設定します。
※入出力クラスタ対応版です。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I32 handle ハンドルを保持する領域を確保します。

I16 NodeAdr ノードアドレスの指定です。設定するノードの番号は１からです。
TCP/IPで複数のノードを接続するときはポート識別番号として使用します。

I16 ErrChk ＣＲＣ／ＬＲＣチェック　有無フラグ
０の時はエラーチェックを行いません。
エラーチェックを行う時は０以外の値を設定してください。

U32 ModelNo モデル番号の配列です。
使用するモジュール番号を設定してください。

- U32 **モジュール型番** 使用するモジュールの型番号

入力ｸﾗｽﾀ ハンドルやノードアドレス、及びエラー情報を保持します。
内部で作成する為、結線は不要です。

- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

I32 Return Wago_IO.dllの WAGO_SetConfig() の戻り値です。
０の場合は正常動作を表し、負の値の場合はエラーを表します。

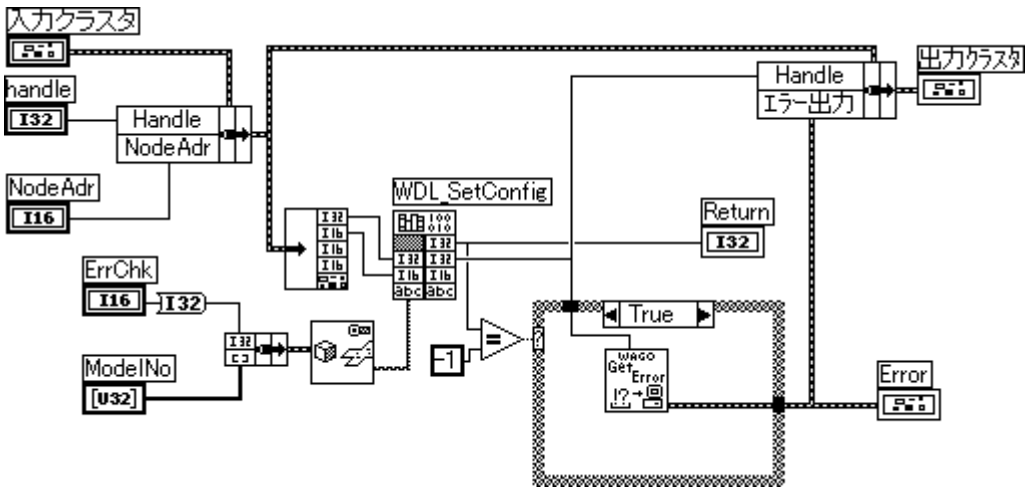
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
処理を行った結果が格納され、次の関数に引き渡されます。

- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

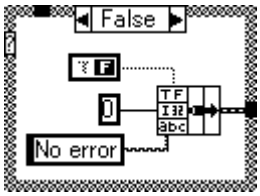
Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

- TF status ｴﾗｰがある場合にはTRUEになります。
- I32 code ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラーが発生した場合はエラーコードを設定し、正常にオープンできた場合はエラー情報のクリアを行います。



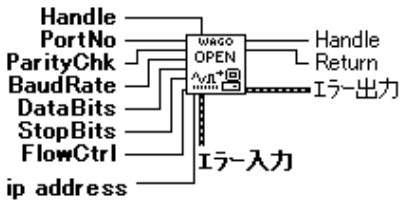
７．ポートオープン（クラスタ対応版）

関数名：WagoOpenComPort.vi

機能：バスカップラとの通信を行う為にシリアルポートをオープンします。

通信を行うポートと通信方式を指定して呼び出します。
※入出力クラスタ対応版です。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

I16 ParityChk シリアル通信時に行うパリティチェックを指定します。
０：無し
１：偶数パリティ
２：奇数パリティ

I16 PortNo オープンするComポートの番号を指定します。

abc ip address TCP/IPで接続するとき、IPアドレスを指定します。

I32 BaudRate 通信速度(ボーレート)を指定します。
9600,19200 等の値を設定します。

I16 DataBits データビット長を指定します。
８：８ビット長
７：７ビット長

I16 StopBits ストップビット長を指定します。

I16 FlowCtrl フローコントロールの指定を行います。
０：無し　　１：Xon/Xoff　　２：ハードウェア

入力ｸﾗｽﾀ ハンドルやノードアドレス、及びエラー情報を保持します。

- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

通信設定 設定された通信設定をデータソースとして引き渡します。

I32 **Return** Wago_IO.dllの WAGO_OpenCommPort() の戻り値です。
0の場合は正常動作を表し、負の値の場合はエラーを表します。

出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
処理を行った結果が格納され、次の関数に引き渡されます。

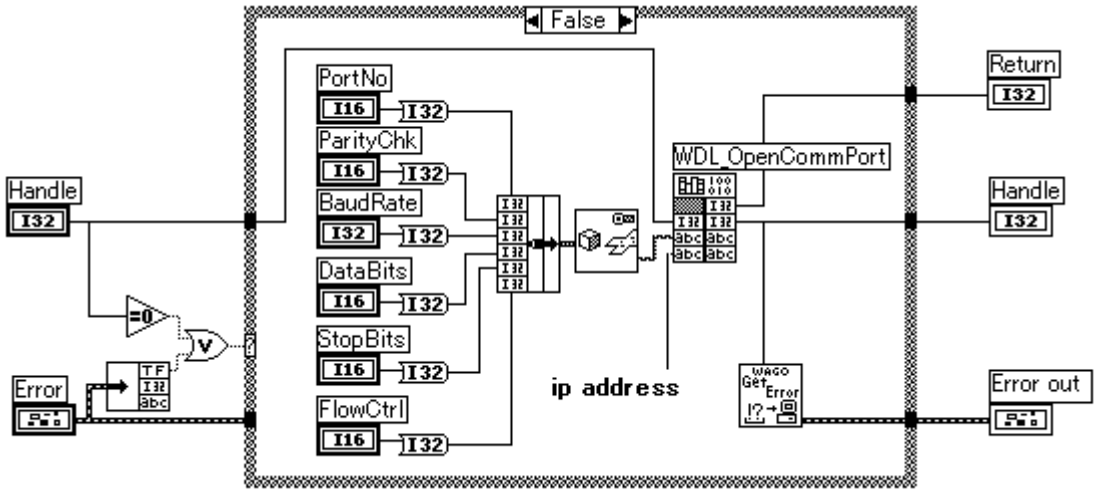
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。

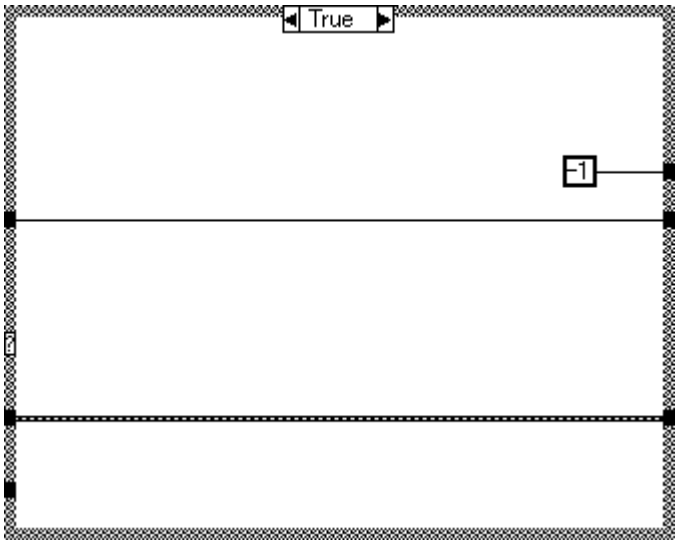
- TF status ｴﾗｰがある場合にはTRUEになります。
- I32 code ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc source ｴﾗｰがどのVIで起こったかを示します。

通信設定 設定された通信設定を保持します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラーが発生している場合は処理を行わない。



８．クローズ（クラスタ対応版）

関数名：WagoClosePort.vi

機能：シリアルポートをクローズします。

※入出力クラスタ対応版です。

ｺﾈｸﾀﾍﾟｰﾝ

制御器と表示器

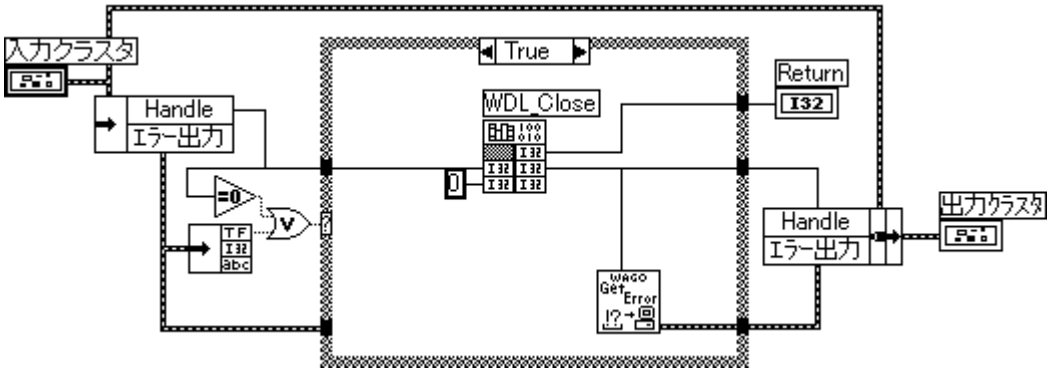
入力ｸﾗｽﾀ ハンドルやノードアドレス、及びエラー情報を保持します。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

I32 **Return** Wago_IO.dllの WAGO_CloseCommPort() の戻り値です。
０の場合は正常終了です。

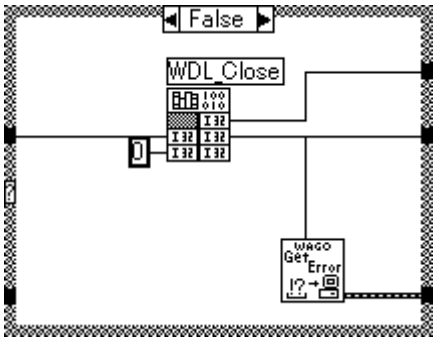
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- **ｴﾗｰ出力**

Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。
- TF **Status** ｴﾗｰがある場合にはTRUEになります。
- I32 **code** ｴﾗｰの種類を示す数値です。0 はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc **source** ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



現在はエラー発生時・正常時も同じ処理



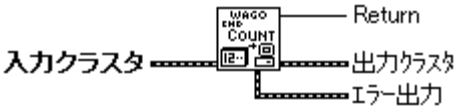
９．コマンド回数確認のためのVI（クラスタ対応版）

関数名：WagoGetCmdCount.vi

機能： コマンド回数確認
指定されたノードのバスカップラから実行されたコマンドの回数を読み出します。
（正常に行われたコマンド回数がバスカップラのメモリカウンタにカウントされます。
バスカップラの電源をオフにするとこのメモリカウンタはリセットされます。）

※入出力クラスタ対応版です。

ｺﾈｸﾀﾍﾟｰｼﾞ



制御器と表示器

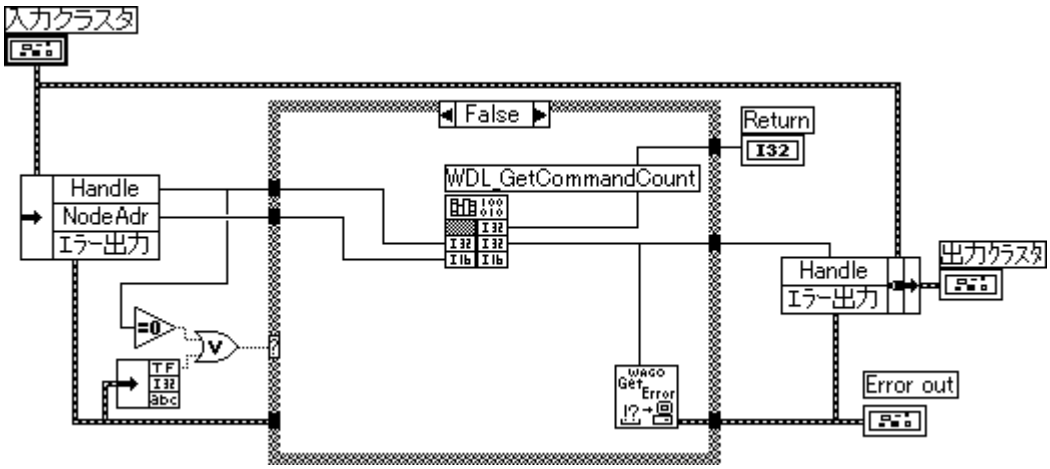
入力ｸﾗｽﾀ ハンドルやノードアドレス、及びエラー情報を保持します。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- ｴﾗｰ出力

I32 **Return** Wago_IO.dllの WAGO_GetCommandCount() の戻り値です。
正の値または０の場合は受け取ったコマンドの数を表します。
負の値の場合はエラーを表します。

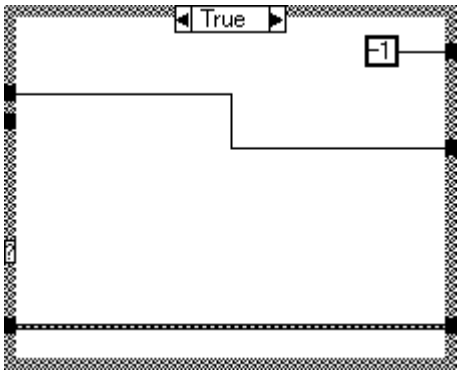
出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。
- I32 Handle
- I16 NodeAdr
- I16 MdlNo
- I16 CnlNo
- ｴﾗｰ出力

Error out もしｴﾗｰ入力にｴﾗｰがあれば、ｴﾗｰ出力にはｴﾗｰ入力と同じ情報が表示されます。それ以外はこのVI内で起こったｴﾗｰを示します。
- TF Status ｴﾗｰがある場合にはTRUEになります。
- I32 code ｴﾗｰの種類を示す数値です。０はｴﾗｰなし、正の数は警告、負の数はｴﾗｰを示します。
- abc source ｴﾗｰがどのVIで起こったかを示します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラー発生時には何もしない。



12.4　サンプルライブラリ解説

本サンプルライブラリは、入出力クラスタを使用したWagoIOデータ入出力用VIであり、目的別に入出力を行うサンプルプログラムを集めたものです。この節ではその使用方法とブロックダイアグラムを述べます。

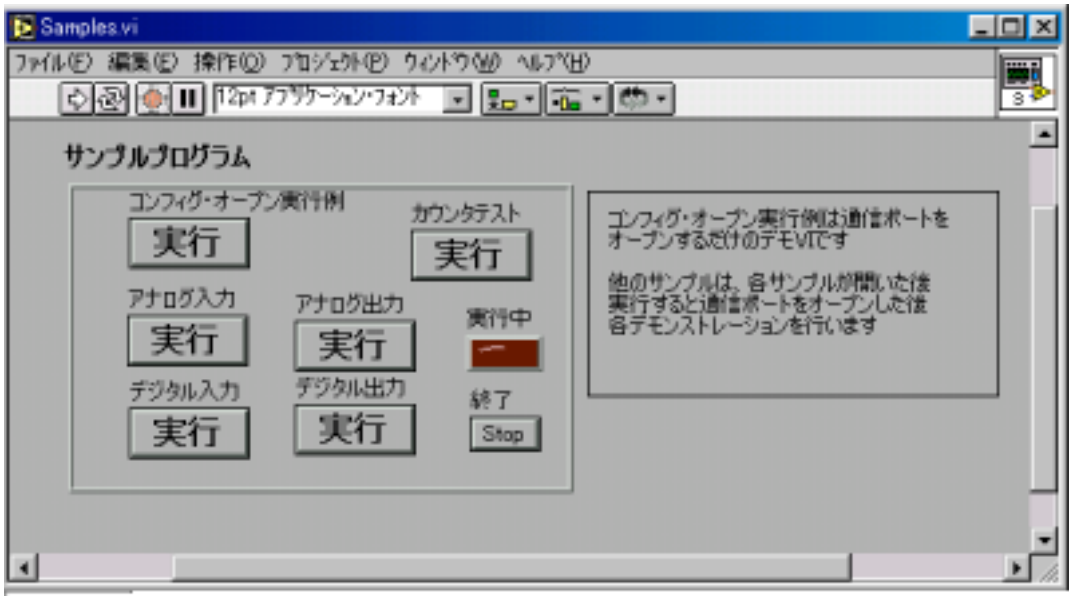
以下のサンプルプログラムがProgramFiles¥Wago_io¥WdllCall¥の下にあるWagoSamples.llbに収録されています。下記のプログラムはWagoSamples.llb内で直接これを選択するか、ProgramFiles¥Wago_io¥WdllCall¥の下にあるSamples.viを開くことによりアクセスすることができます。

サンプル名：WagoPnlConfig.vi	入出力前にノードを設定し、通信ポートをオープンします。
サンプル名：Wago_AI_Sample.vi	アナログ値を取得し、波形チャートに表示します。
サンプル名：Wago_AO_Sample.vi	スライダで指定したアナログの値をIOモジュールへ出力します。
サンプル名：Wago_DI_Sample.vi	デジタル値を取得し、LED配列で表示します。
サンプル名：Wago_DO_Sample.vi	スイッチ配列で指定したデジタル値をIOモジュールに出力します。

このうち、**WagoPnlConfig.vi**は、操作用VIに含まれるべき物ですが、フロントパネルを開き値を設定する物なので、ユーザーのデザイン変更に対応する為、サンプルとしています。パネルデザインや色などをカスタマイズして使用してください。

WagoPnlConfig.vi の出力ｸﾗｽﾀで直接入出力アクセスを行うことが出来ます。

Samples.viを開くと次のような画面が表れます。



操作対象となる機能に対応した「実行」ボタンを押すとそのプログラムの動作が開始します。

12.4.1　　サンプル名：WagoPnlConfig.vi

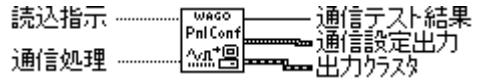
機能：　コンフィグレーションとシリアルポートのオープンを行います。
各設定値を設定して「実行」ボタンを押すと、処理を行います。

正常にオープンが行われた場合は、フロントパネルが閉じ、次の処理に移ります。
Samples.Vi の状態では、すぐにクローズして終了します。

エラーが発生した場合は、ダイアログが表示されます。
継続を選ぶことで、エラークリア等の処理が行えます。

設定された内容はファイルに保存したり、読み出すことが可能です。
端子を接続し、True値を与えることで、自動実行やファイルからのデータ取り込みを指示できます。

ｺﾈｸﾀﾍﾟｰﾝ



制御器と表示器

TF　**読込指示** Trueにしておくと、自動でファイルから設定を読み込みます。

TF　**通信処理** 設定を終了し、現在の設定でポートのオープンを行います。
接続した場合は入力待ちにならずに実行します。

I32　**通信テスト結果** Wago_IO.dllの WAGO_GetCommandCount() の戻り値です。 正の値または０の場合は受け取ったコマンドの数を表します。 負の値の場合はエラーを表します。 ここでは初回の通信を行い、応答があるかどうかを見ています

通信設定出力 設定された通信設定を保持します。

出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。

設定パラメータ

I16　NodeAdr ノードアドレスの指定です。 設定するノードの番号は１からです。

I16　ErrChk ＣＲＣ／ＬＲＣチェック　有無フラグ ０の時はエラーチェックを行いません。
エラーチェックを行う時は０以外の値を設定してください。

U32　ModelNo モデル番号の配列です。 使用するモジュール番号を設定してください。

I32　BaudRate 通信速度(ボーレート)を指定します。 9600,19200 等の値を設定します。

I16　PortNo オープンするComポートの番号を指定します。

I16　ParityChk シリアル通信時に行うパリティチェックを指定。 ０：無し １：偶数パリティ ２：奇数パリティ

I16 FlowCtrl フローコントロールの指定を行います。０：無し １：Xon/Xoff ２：ハードウェア

I16 StopBits ストップビット長を指定します。

I16 DataBits データビット長を指定します。８：８ビット長 ７：７ビット長

abc ip address TCP/IP接続する場合のIPアドレスを指定します。

コマンドボタン

TF **読込指示** 設定ファイルに記述された内容をフロントパネルに設定します。

TF **書込指示** 現在の設定内容を設定ファイルに書き出します。

TF **エラー情報** エラーメッセージを消去します。

TF **通信処理** 設定を終了し、現在の設定でポートのオープンを行います。

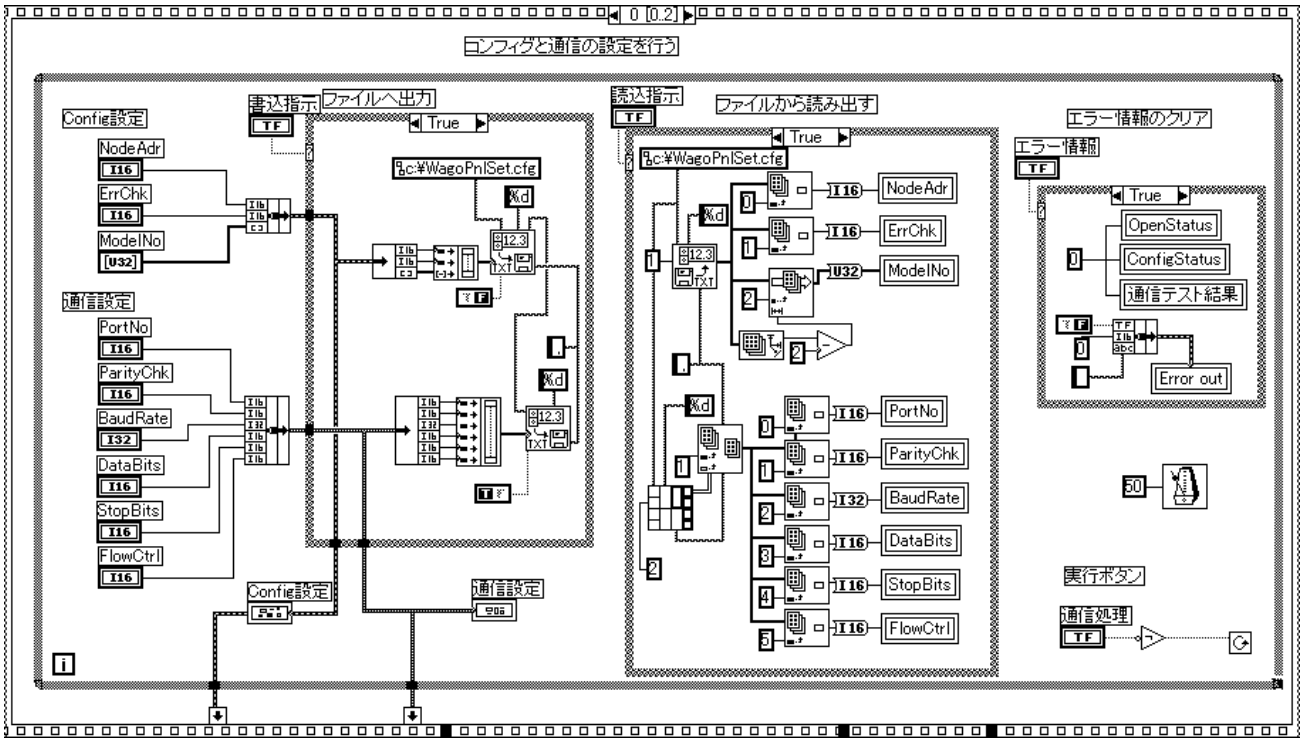
ステータス他

I32 ConfigStatus コンフィグが正常に行われたかどうかの確認用です。 データ収集動作には不要です。

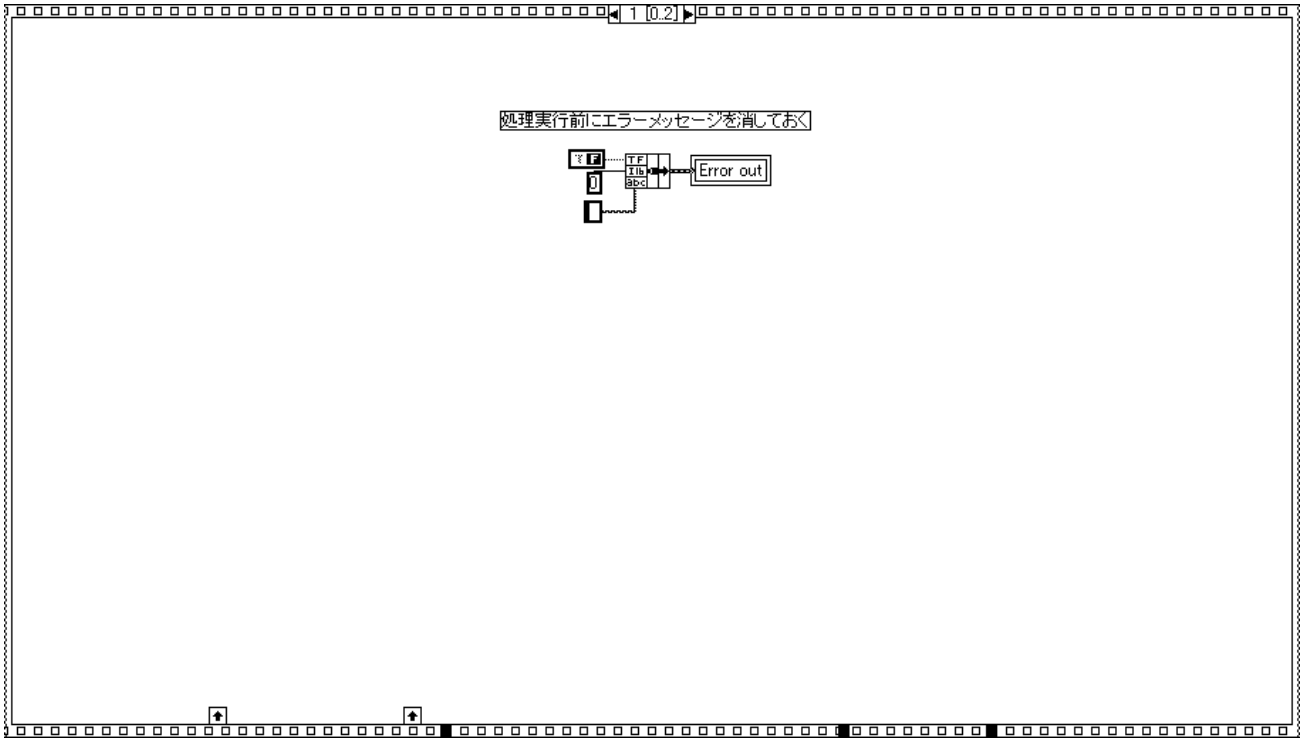
I32 OpenStatus シリアルポートが正常にオープンできたかの確認用です。 データ収集動作には不要です。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ

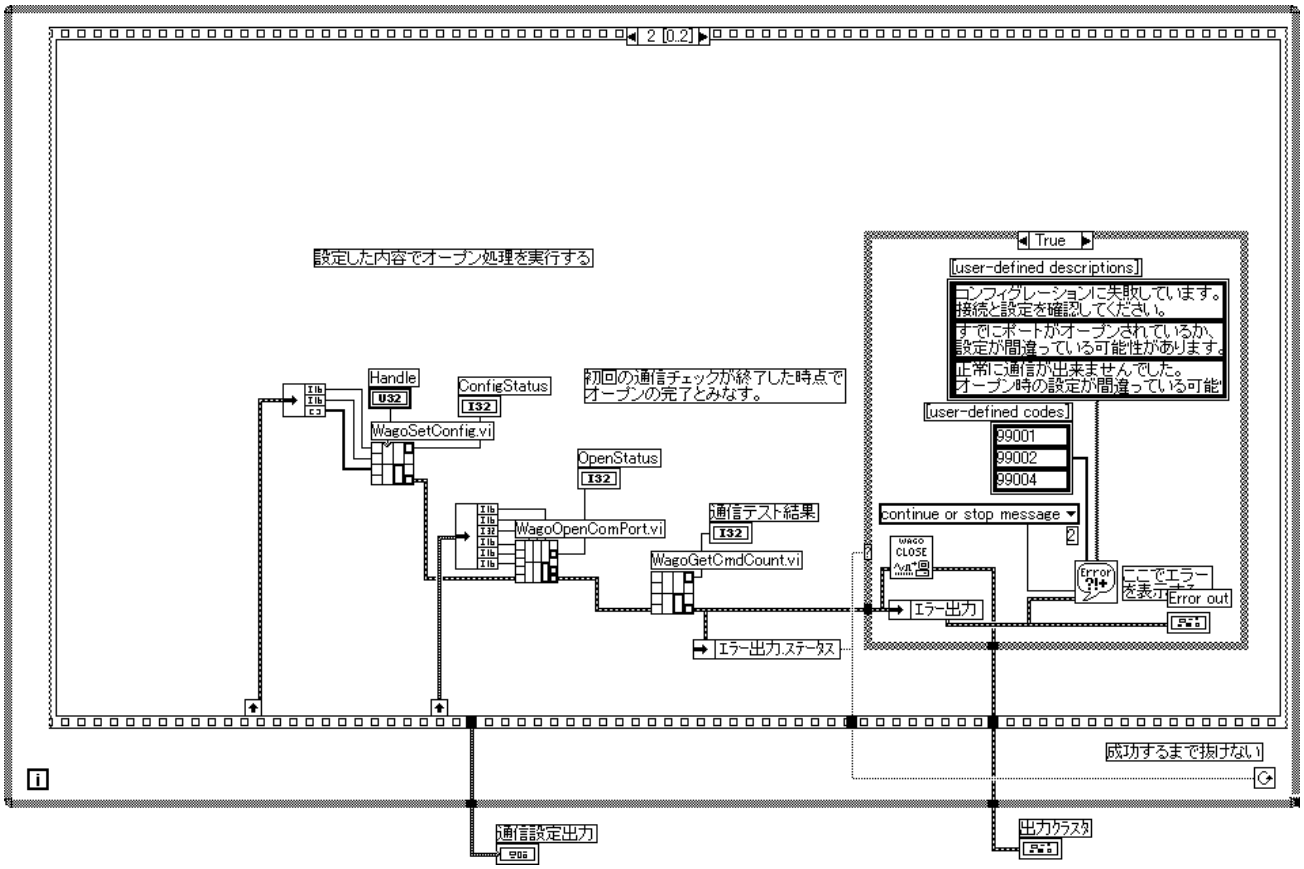
設定パート



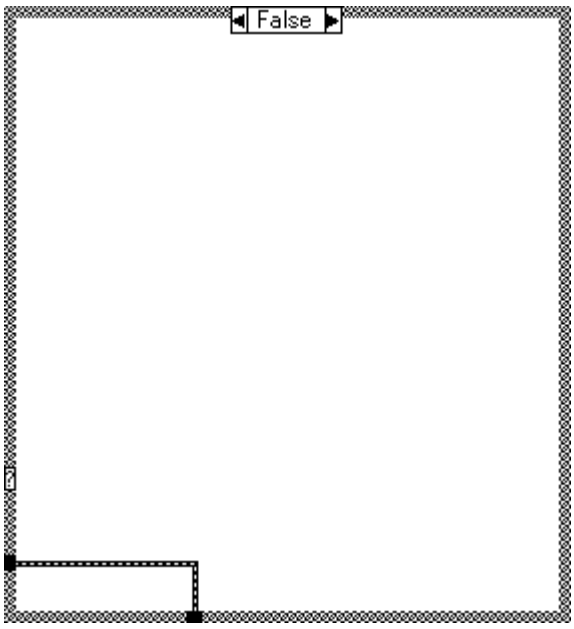
開始前にエラー情報をクリア



初回の通信テスト



エラーが発生しない時はそのままのデータを送って抜ける。



12.4.2 サンプル名：Wago_AI_Sample.vi

機能： アナログ入力のサンプルプログラムです。
モジュール番号、チャンネル番号、取得チャンネル数を指定して実行してください。

ｺﾈｸﾀﾍﾟｰﾝ

制御器と表示器

TF **読込指示** Trueにしておくと、自動でファイルから設定を読み込みます。

出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。

設定パラメータ

I16 **チャンネル番号** モジュール内のチャンネルを指定します。

I16 **モジュール番号** モジュールの番号を指定します。

I32 **チャンネル数** 操作を行うチャンネルの数を指定します。

コマンドボタン

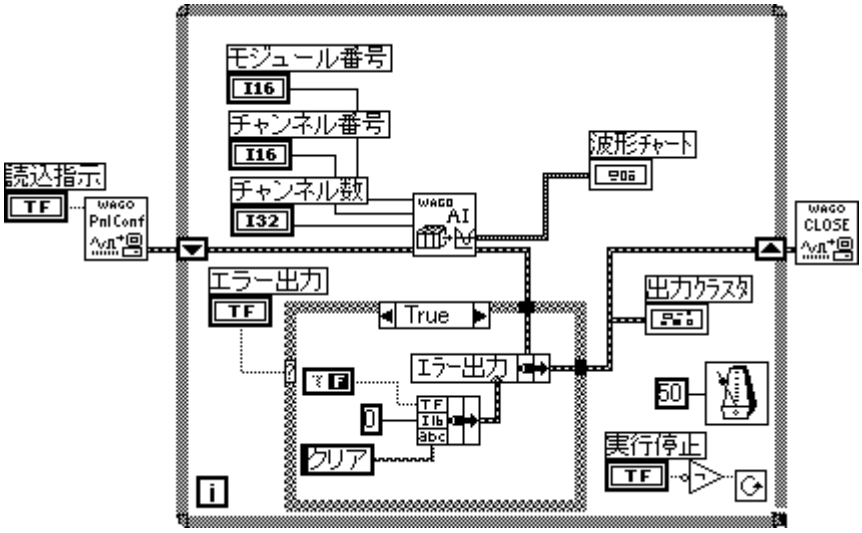
TF **エラー出力** 通信エラー情報をクリアします。

TF **実行停止** 押すとVIが終了します。

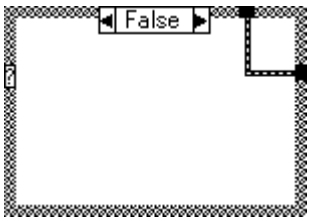
表示内容

波形ﾁｬｰﾄ 取得したアナログデータを波形チャートとして確認することが出来ます。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラークリアを押さないときはそのままのデータ



12.4.3 サンプル名：Wago_AO_Sample.vi

機能： アナログ出力のサンプルプログラムです。
モジュール番号、チャンネル番号を指定して、出力するアナログデータ（スライダ）を操作してください。

ｺﾈｸﾀﾍﾟｰﾝ

制御器と表示器

TF **読込指示** Trueにしておくと、自動でファイルから設定を読み込みます。

出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。

設定パラメータ

I16 **チャンネル番号** モジュール内のチャンネルを指定します。

I16 **モジュール番号** モジュールの番号を指定します。

I32 **チャンネル数** 操作を行うチャンネルの数を指定します。

コマンドボタン

TF **エラー出力** 通信エラー情報をクリアします。

TF **実行停止** 押すとVIが終了します。

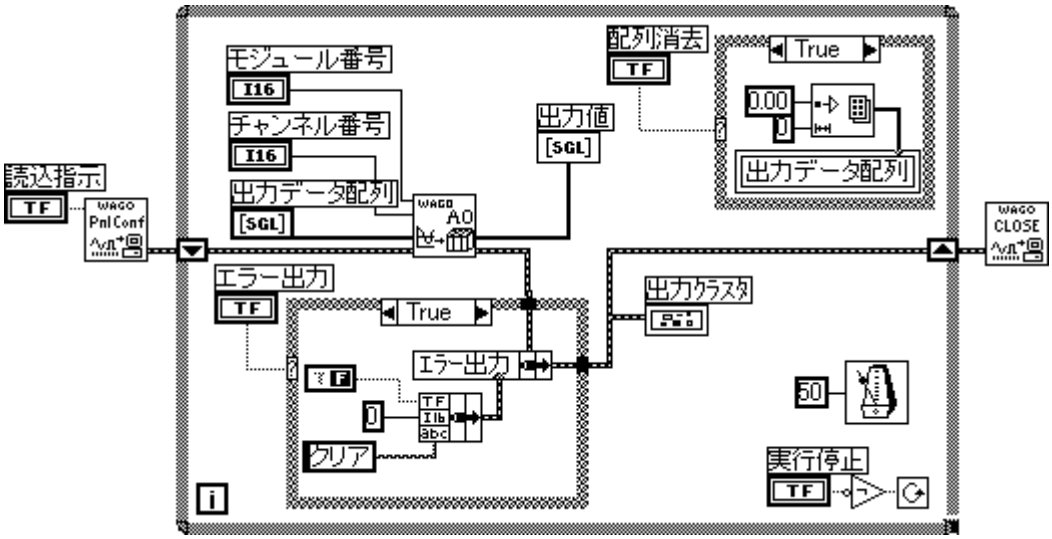
TF **配列消去** 配列の領域を消去します チャンネル数を減らす時に使用してください。

表示内容

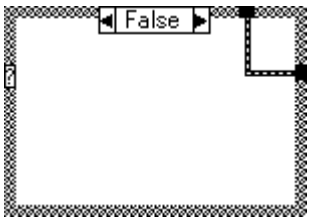
SGL **ｽﾗｲﾄﾞ** 出力するアナログデータの値をスライダで設定します。

[SGL] **出力値** 出力した値を数値で表示します。

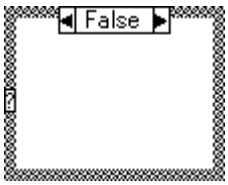
ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラークリアを押さないときはそのままのデータ



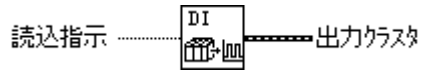
配列消去を指定しなければ通常動作です。



12.4.4 サンプル名：Wago_DI_Sample.vi

機能： デジタル入力のサンプルプログラムです。
モジュール番号、チャンネル番号、取得チャンネル数を指定して実行してください。

ｺﾈｸﾀﾍﾟｰｼﾞ



制御器と表示器

TF **読込指示** Trueにしておくと、自動でファイルから設定を読み込みます。

出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。

設定パラメータ

I16 **チャンネル番号** モジュール内のチャンネルを指定します。

I16 **モジュール番号** モジュールの番号を指定します。

I32 **チャンネル数** 操作を行うチャンネルの数を指定します。

コマンドボタン

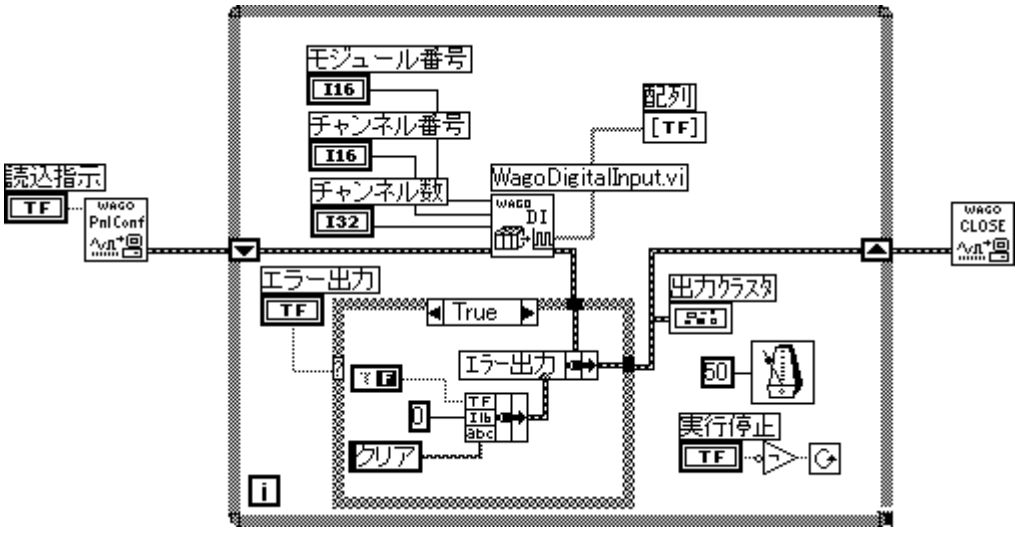
TF **エラー出力** 通信エラー情報をクリアします。

TF **実行停止** 押すとVIが終了します。

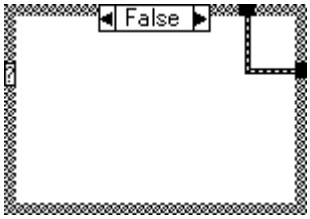
表示内容

[TF] **配列** 読み出したデータです。Onになっているデータが点灯します。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラークリアを押さないときはそのままのデータ



12.4.5 サンプル名：Wago_DO_Sample.vi

機能： デジタル出力のサンプルプログラムです。
モジュール番号、チャンネル番号、設定チャンネル数を指定して実行してください。

ｺﾈｸﾀﾍﾟｰﾝ

制御器と表示器

TF **読込指示** Trueにしておくと、自動でファイルから設定を読み込みます。

出力ｸﾗｽﾀ Wago_ioモジュールとのやり取りのために必要な情報を保持し、処理の順番を決定します。

設定パラメータ

I16 **チャンネル番号** モジュール内のチャンネルを指定します。

I16 **モジュール番号** モジュールの番号を指定します。

I32 **チャンネル数** 操作を行うチャンネルの数を指定します。

コマンドボタン

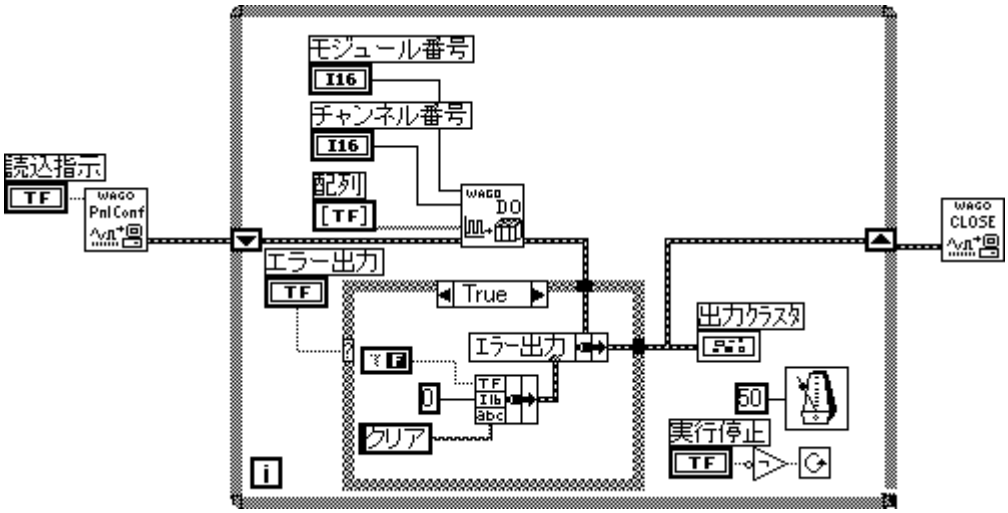
TF **エラー出力** 通信エラー情報をクリアします。

TF **実行停止** 押すとVIが終了します。

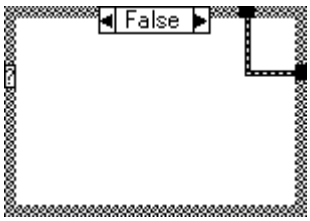
表示内容

[TF] **配列** 出力する値を設定します。 Onにするチャンネルの**LED**を点灯させてください。

ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ



エラークリアを押さないときはそのままのデータ



12.5　データ収集アプリケーション例　（DAQ_Apply.vi）

データ収集を行い、ログファイルとして収集するプログラムです。

機能

収集周期、収集チャンネル数の変更が行えます。
実行すると、先ずコンフィグ・通信の設定画面が開きますので、そこでモジュール構成、通信設定を行い、実行ボタンを押してください。
通信が成功すると、「待機中」になり、ステータスランプが点灯します。
ここで取得するモジュール、チャンネル、チャンネル数、収集周期などを設定して、「データ記録開始/停止」ボタンを押してください。データの収集が開始されます。

ログファイル記録

最初にログファイルの初期化が必要です。
ログファイルが初期化されると、以降のログファイルへの出力は、全て初期化されたファイルに対して行われます。

ログファイルに保存するには、「ログファイル操作」の指定を変更します。
・毎回保存　　一回ごとに収集したデータをファイルに追加していきます。
・蓄積保存　　収集したデータを蓄積領域に追加していきます。ファイルの保存は「保存指示」にて行われます。
出力されるファイルは、csv形式です（Excel等に直接読み込めます）。

実行終了ボタンを押すと、処理を終了します。

制御器と表示器

I16　**モジュール番号**　操作を行うモジュールの番号を指定します。

I16　**チャンネル番号**　操作を行う開始チャンネルを指定します。

I32　**チャンネル数**　データを取得するチャンネル数を指定します。

TF　**エラークリア**　チャンネル設定時などに起きたエラーのメッセージを消去します。

U32　**収集周期(ms)**　サンプリング周期を指定します（単位はｍｓ）。

TF　**データ記録開始/停止**　データ収集を開始します。

U16　**ログファイル操作**　ログファイルの操作を指定します。

TF　**ファイル作成**　ログファイルを初期化します。

TF　**保存指示**　初期化されているログファイルに、蓄積されたデータを出力します。

TF　**実行終了**　DAQアプリケーションを終了します。

表示内容

	通信ステータス	通信時のエラーメッセージを表示します。
	収集データ	LabVIEWの標準部品（波形チャート）VIを使用しています。 グラフの色以外は標準値を使用しておりますので、目的に応じて変更してください。

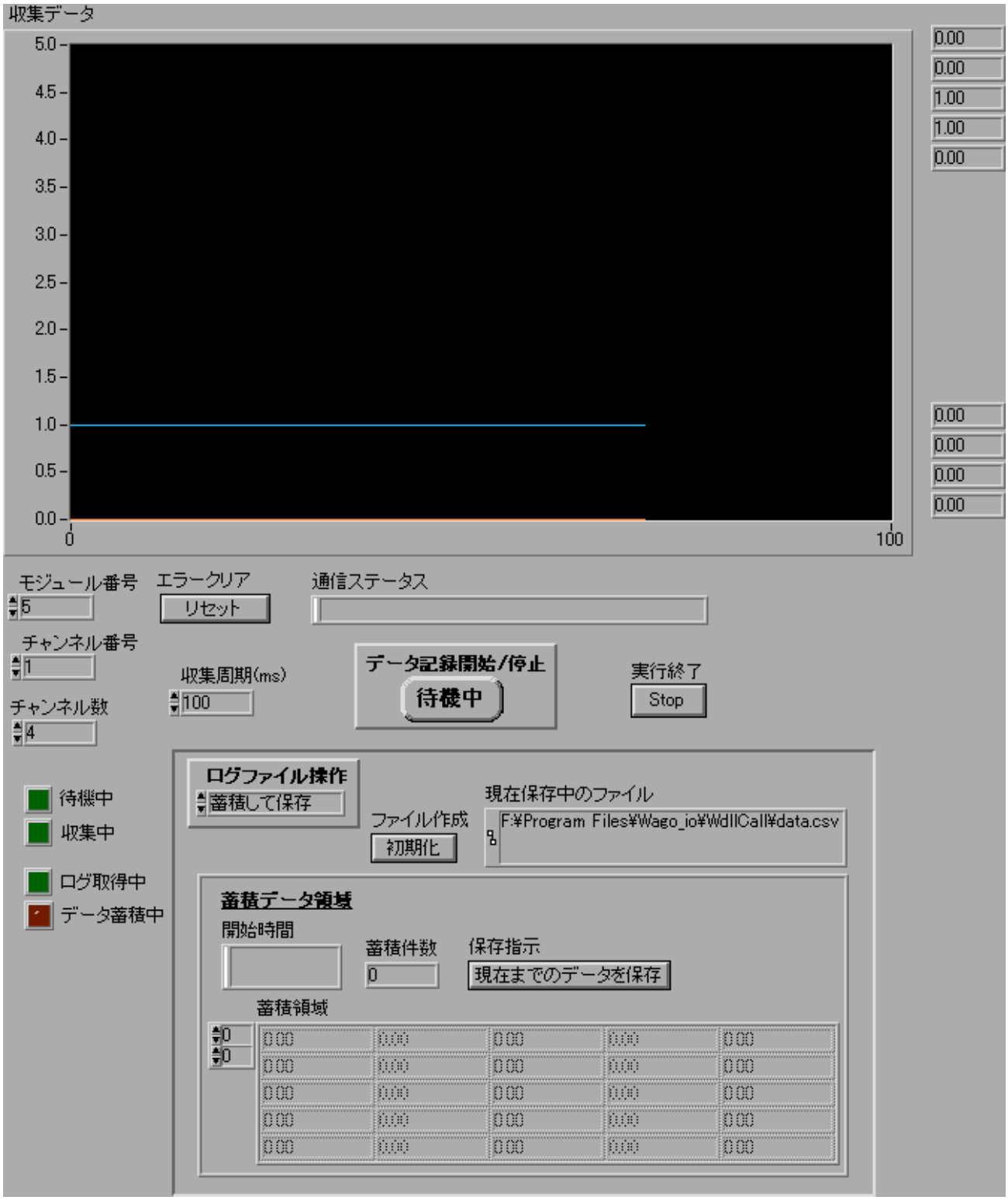
インジケータ

	待機中	データ収集を行っていない時に点灯します。
	収集中	データ収集を行っている時に点灯します。
	ログ取得中	ログ収集をしながらデータを取り込んでいる時に点灯します。
	データ蓄積中	ログ収集のうち、データ蓄積を行っている時に点灯します。

ログファイル関連

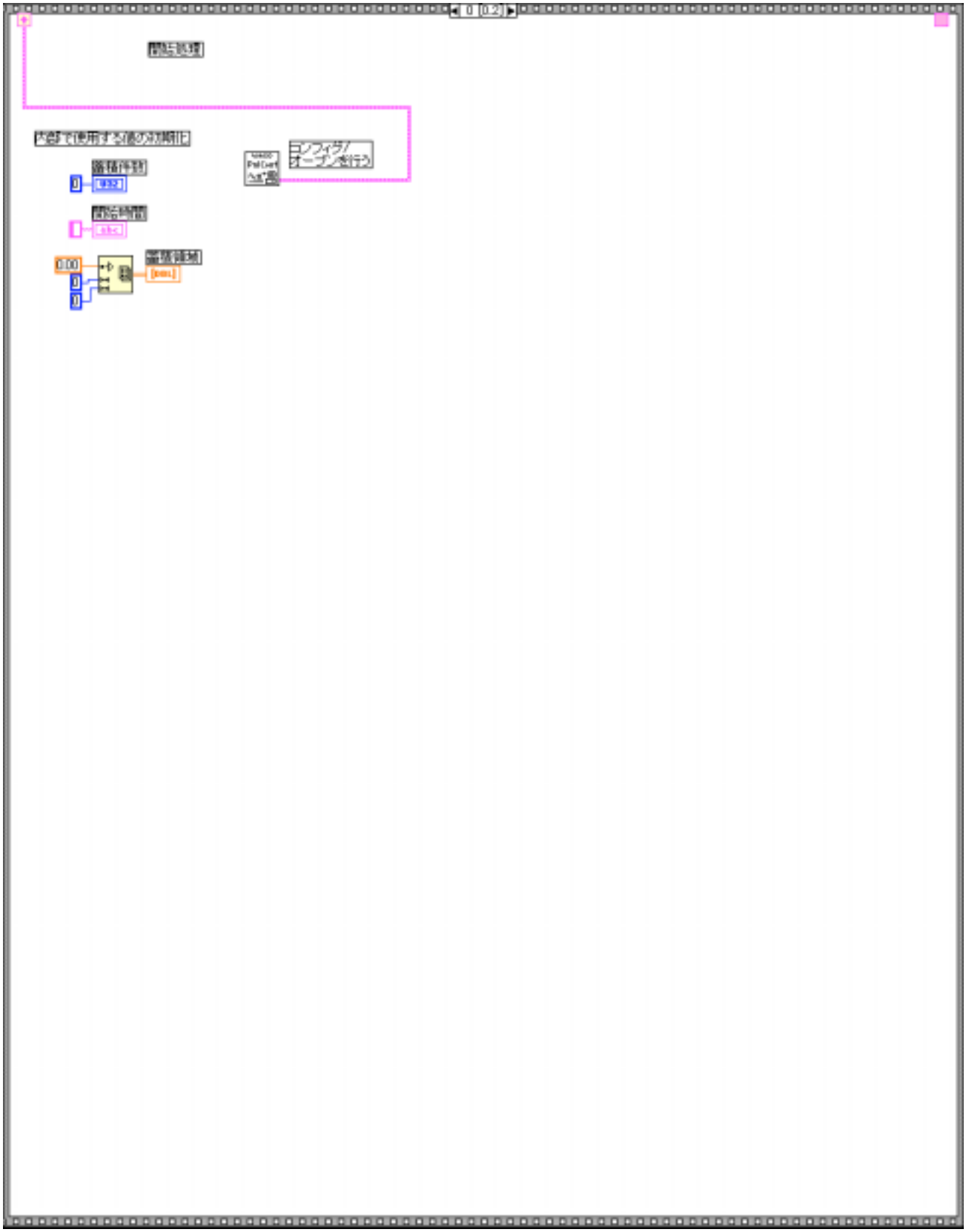
	現在保存中のファイル	ログデータを保存するファイル名です。
	開始時間	データの蓄積を始めた時間です。
	蓄積件数	現在蓄積されているデータの件数。
	蓄積領域	現在蓄積されているデータの領域。

フロントパネル


収集データ
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
0.00
0.00
1.00
1.00
0.00
0.00
0.00
0.00
0.00
モジュール番号
5
エラークリア
リセット
通信ステータス
チャンネル番号
1
収集周期(ms)
100
チャンネル数
4
データ記録開始/停止
待機中
実行終了
Stop
待機中
収集中
ログ取得中
データ蓄積中
ログファイル操作
蓄積して保存
ファイル作成
初期化
現在保存中のファイル
F:¥Program Files¥Wago_io¥WdllCall¥data.csv
蓄積データ領域
開始時間
蓄積件数
0
保存指示
現在までのデータを保存
蓄積領域

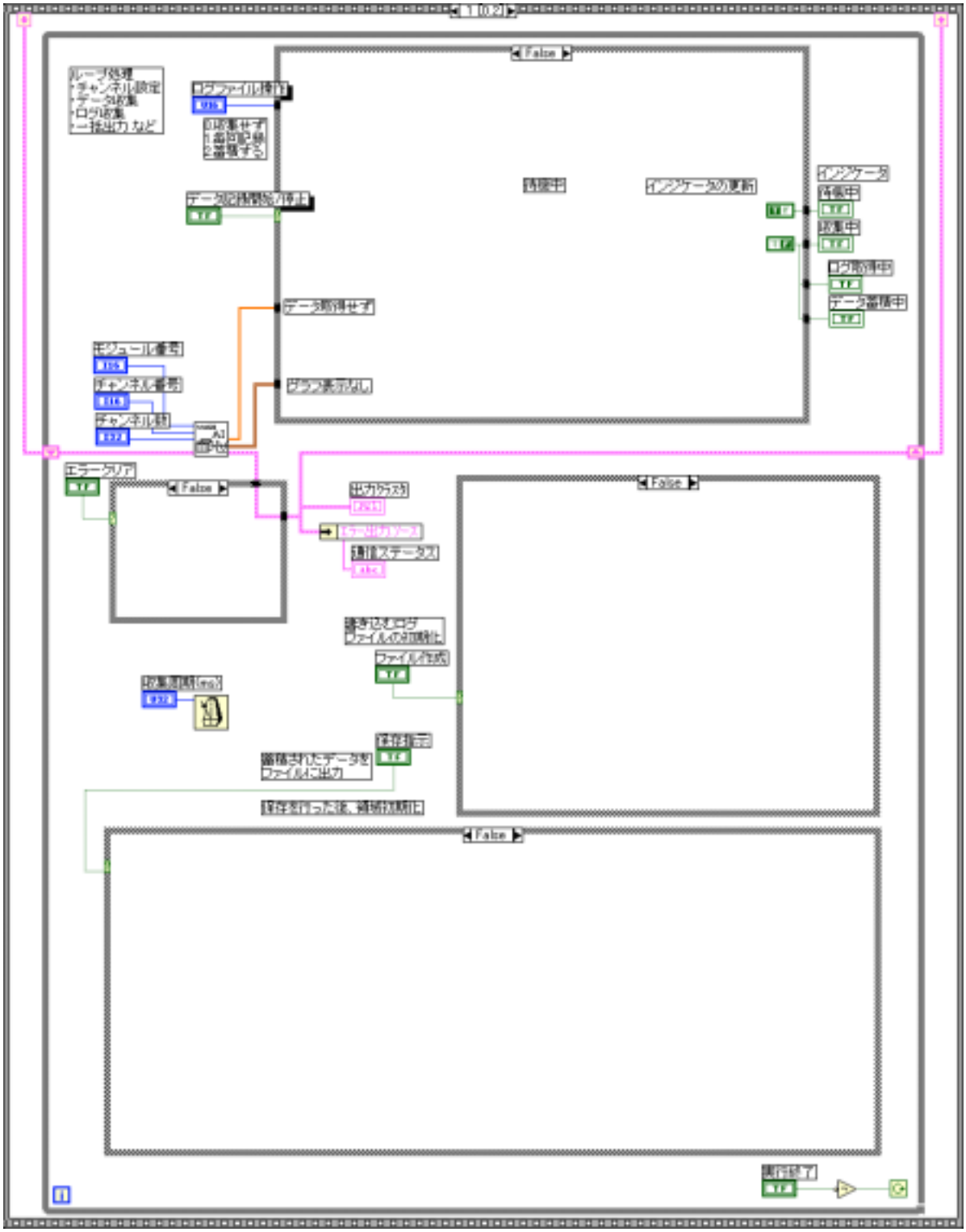
ﾌﾞﾛｯｸﾀﾞｲｱｸﾞﾗﾑ

開始処理

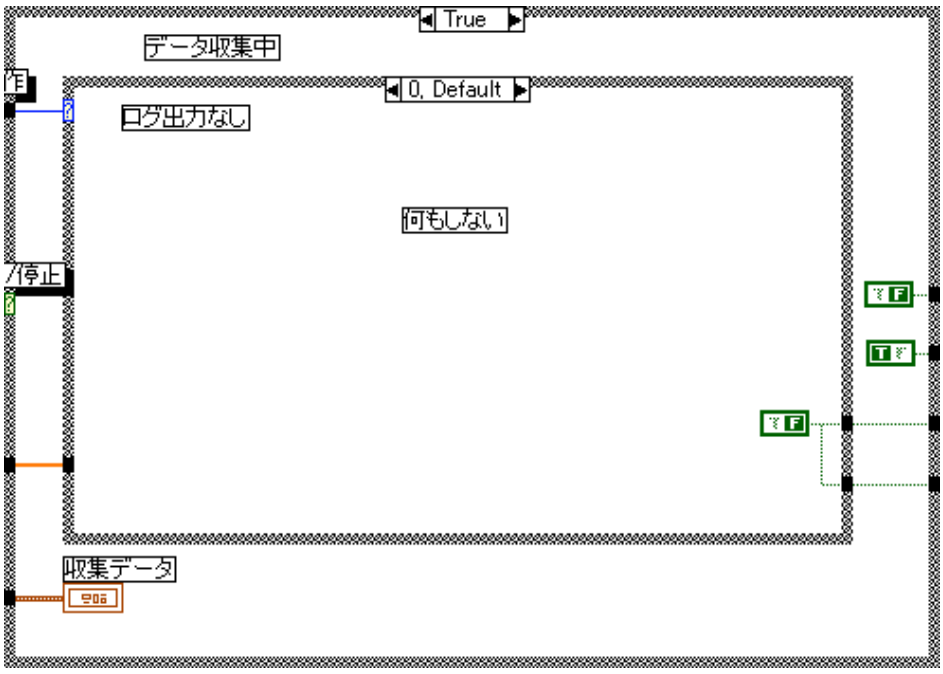


開始処理では、変数の初期化と、通信ポートのオープンを行っています。

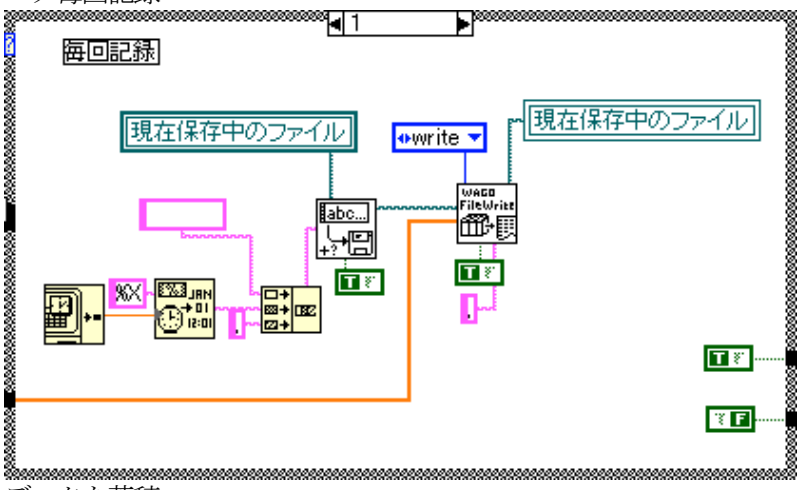
データ収集待機中


False
ループ処理
・チャンネル設定
・データ収集
・ログ収集
・一括出力 など
ログファイル指定
収集停止
1:巻回記録
2:巻積する
データ記録開始の待機
待機中
インジケータの更新
インジケータ
待機中
記録中
ログ書込中
データ蓄積中
データ取得せず
グラフ表示なし
モジュール番号
チャンネル番号
チャンネル数
エラークリア
False
出力クラス
通信ステータス
False
書き込むログファイルの初期化
ファイル作成
収集周期[ms]
蓄積されたデータをファイルに出力
保存を行った後、蓄積初期化
False
実行終了

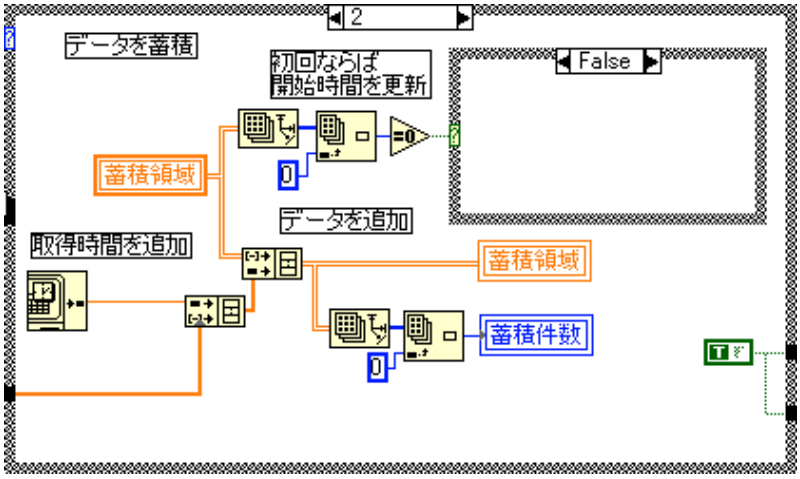
データ収集中
（ログ無し）

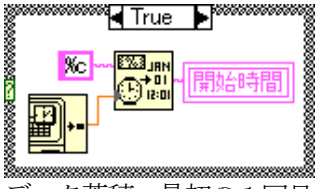


ログ毎回記録



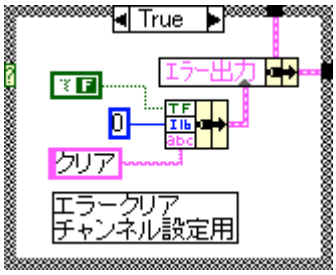
データを蓄積



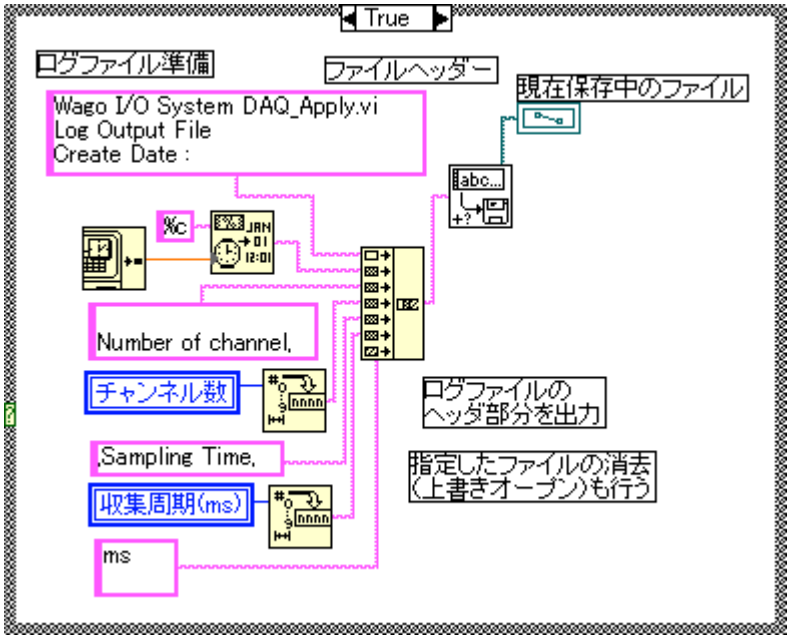


データ蓄積。最初の１回目

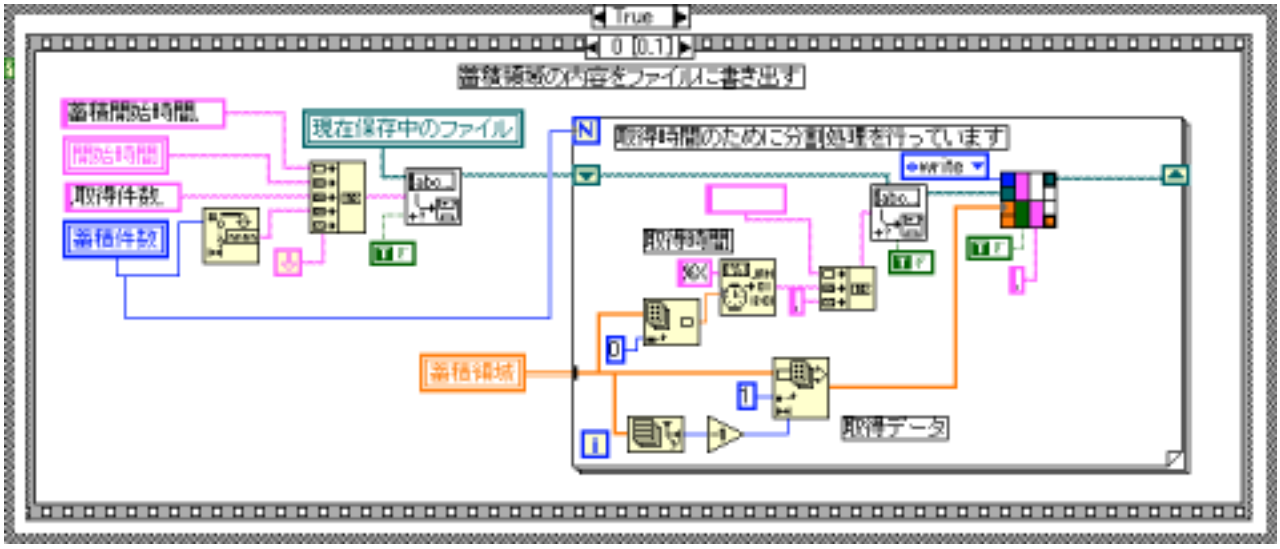
エラーメッセージ（通信ステータス）をクリア



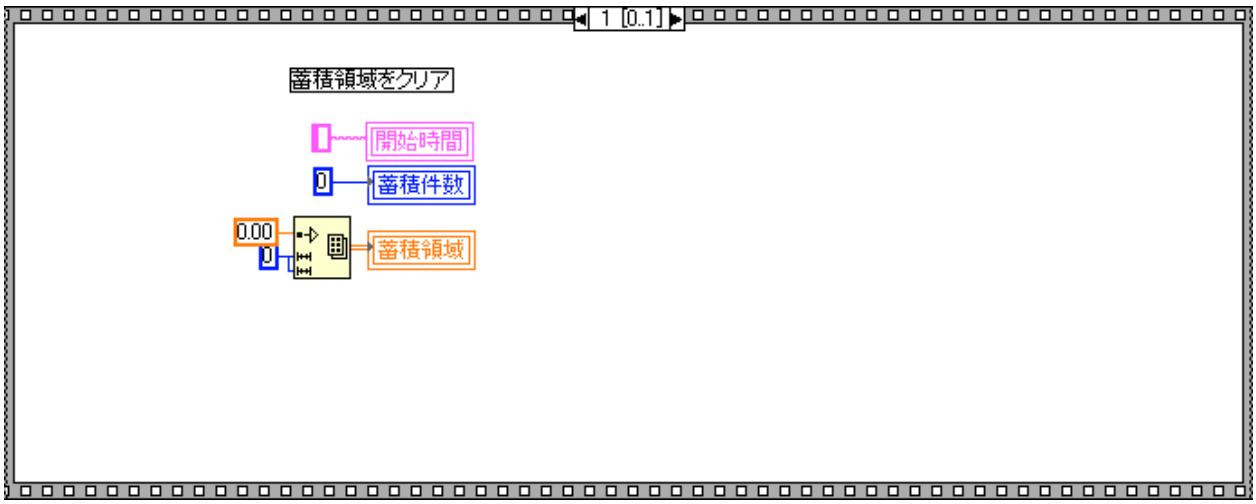
ログファイルの初期化を行ったとき



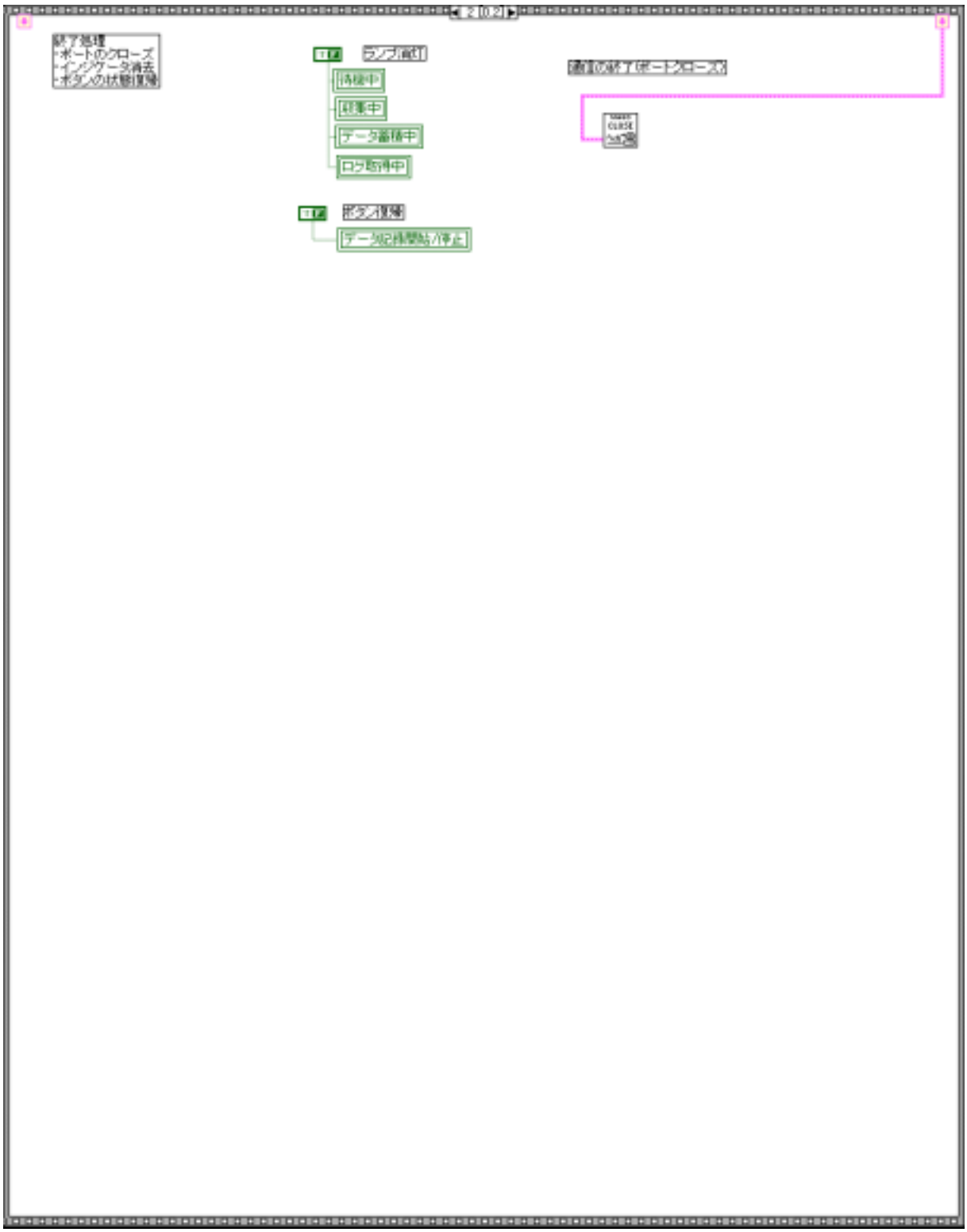
蓄積データの出力を押した時
１．データを出力



２．出力した領域をクリア



終了処理
ポートのクローズとインジケータの消灯



13 LabVIEW 実行ソフトウェアサンプル

ここではLabVIEW を持っていないユーザーのために、ワゴジャパン㈱にて作成された実行プログラムのサンプルを紹介いたします。このサンプルプログラムは LabVIEW 開発システムを用いて作られたものですが、LabVIEW 開発ツールを持っていなくても利用することができます。

＜ソフトウェアの種類：2002 年 7 月 1 日現在＞

ソフトウェア名	最大測定 Ch 数	リアルタイム平均記録	ﾚﾝｼﾞﾄﾘｶﾞ、ﾀﾞｲﾅﾐｯｸﾄﾘｶﾞ	警報出力 (4ﾋﾞｯﾄ分)	タイマ繰り返し計測
DAQ_Sample1	8	○	○		
DAQ_Sample2	32 (任意 8Ch 表示)	○	○		
DAQ_Sample3	32 (連続 8Ch 表示)	○	○		
DAQ_Sample1A	8	○	○	○	
DAQ_Sample2A	32 (任意 8Ch 表示)	○	○	○	
DAQ_Sample3A	32 (連続 8Ch 表示)	○	○	○	
DAQ_Sample1B	8	○		○	○
DAQ_Sample2B	32 (任意 8Ch 表示)	○		○	○
DAQ_Sample3B	32 (連続 8Ch 表示)	○		○	○
DAQ_Sample4	160 4 ﾉｰﾄﾞ接続	○	○		
DAQ_Sample4B	160 4 ﾉｰﾄﾞ接続	○			○

●最新バージョンは　http://www.wago.com/wagoweb/japan/jap/index.htm よりダウンロードしてください。
● レンジトリガ～特定の 1 Ch だけを監視して設定された範囲内にあるときだけ自動的に計測します。
●ダイナミックトリガ～表示されている 8Ch 分データが 1 つでも設定された範囲にあるとき自動的に計測します。
●警報出力を利用する場合は、デジタル出力モジュールと併せてご利用ください。
タイマ繰り返し計測～開始時刻、繰り返し計測時間と繰り返し回数を設定して自動計測します。

13.1　データ収集システム　DAQ_Sample1～４

＜以下に、DAQ_Sample１.exe を例として説明します。＞
DAQ_Sample1.exe　は簡易データ収集システムとしてのサンプルプログラムです。以下の手順でセットアップを行った後起動することができます。Windows /98/NT/2000 上で動作します。**このソフトウェアをインストールする前に必ず DLL をインストールして下さい（取扱説明書第６章を参照のこと）。**

１．セットアップ手順

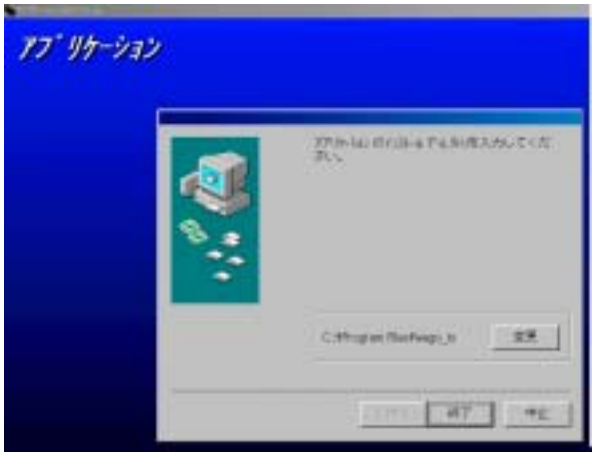
１－１　セットアップは CD-ROM またはホームページからのダウンロードのどちらかを用いて行います。

１）**CD-ROM の場合**
CD-ROM 内に Application_1 のフォルダーがありますのでこのフォルダーを開きます。
このフォルダーの中に disks というサブフォルダーがあり、これを開いた後 Setup.exe をダブルクリックします。以下の画面１－１が表れますので１－２に行きます。

２）**ホームページからのダウンロードの場合**
ホームページから Daq1p.exe を選びダウンロードします。これは自己解凍形式ですのでダブルクリックすると解凍されて Application_1 のフォルダーが作成されます。このフォルダーの下には disks サブフォルダーがあり、その中にある Setup.exe をダブルクリックします。以下の画面１－１が表れますので１－２に行きます。フォルダー番号とプログラムの対応は以下のとおりです。

ソフトウェア名	ホームページ上の圧縮ファイル名	フォルダー名
DAQ_Sample1	Daq1p.exe	Application_1
DAQ_Sample2	Daq2p.exe	Application_2
DAQ_Sample3	Daq3p.exe	Application_3
DAQ_Sample1A	Daq1Ap.exe	Application_1A
DAQ_Sample2A	Daq2Ap.exe	Application_2A
DAQ_Sample3A	Daq3Ap.exe	Application_3A
DAQ_Sample1B	Daq1Bp.exe	Application_1B
DAQ_Sample2B	Daq2Bp.exe	Application_2B
DAQ_Sample3B	Daq3Bp.exe	Application_3B
DAQ_Sample4	Daq4p.exe	Application_4
DAQ_Sample4B	Daq4Bp.exe	Application_4B

画面１－１



１－２　このアプリケーションをインストールするディレクトリを指定します。変更があれば**変更**ボタンを選びます。このままでよければ**終了**ボタンをクリックします。この時点でファイルのインストールが自動的に行われます。この後次の画面が表れますので、**OK** をクリックしインストレーションを終了します。

１－３　設定したパラメータ値保存用ファイルを作成します。

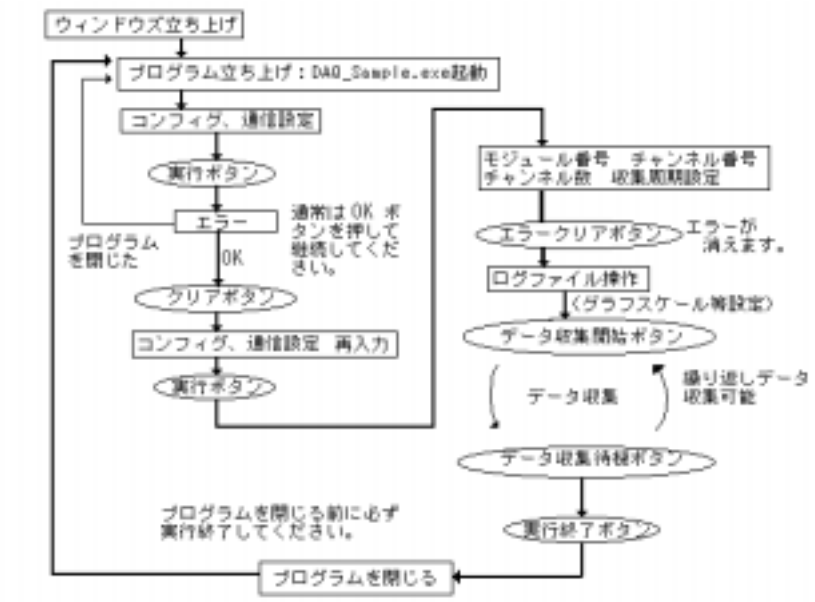
ソフトウェアを最初に起動したとき、設定条件を記憶を２回クリックして条件設定記憶ファイルを作成してください。この動作で条件記憶ファイルが自動的に生成されます。この作業を行わずに設定条件を呼出をクリックしてしまうと、すべてのパラメータが「０」に設定されてしまいます。

条件を記憶するファイルは２つありますので「置き換え」を２回クリックしてください。プログラムメイン画面においても同様の操作で条件記憶ファイルを作成してください。（２－３－３を参照してください。）

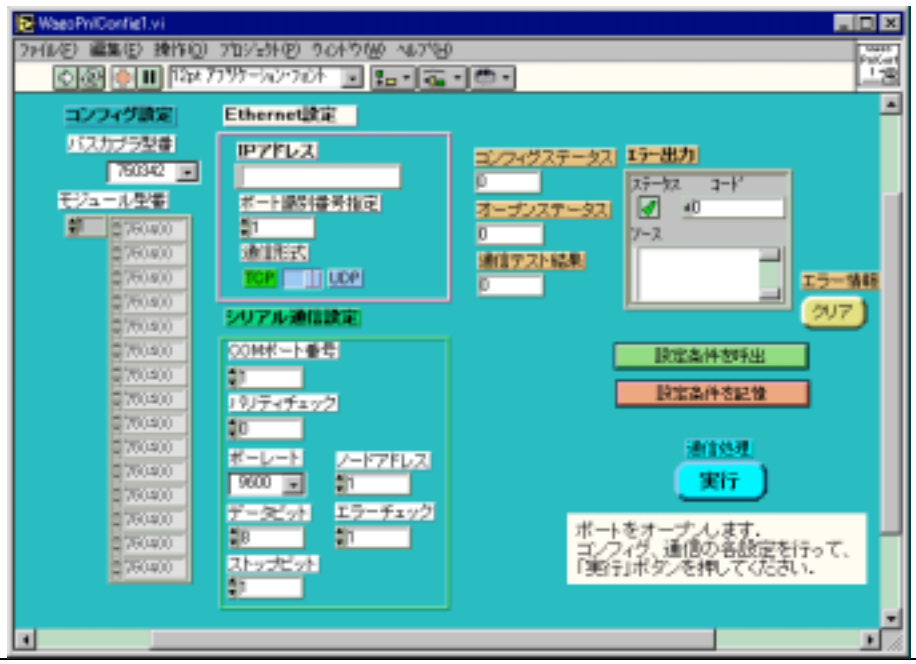
２．DAQ_Sample1.exe の起動

フォルダーProgramFiles/Wago_io の下に **DAQ_Sample1.exe** が作成されていることを確認してください。

２－１　操作の手順



２－２　**DAQ_Sample1.exe** を選択し、ダブルクリックします。以下のように WagoPnlConfig1.vi の画面が表れます。



この画面において初期設定（モジュール構成及び通信パラメータ設定）を行います。途中でプログラムを閉じた場合は、プログラム全体を閉じて、プログラムの起動から再度やり直してください。

３．　コンフィグ及び通信パラメータの設定

画面上のパラメータは PC に接続されたワゴ I/O システムの構成内容に応じて変更します。値の入力は▲または▼をクリックして数字を増減させるか、テンキーから直接数字を入れます。

３－１　コンフィグ（Config）設定

（この設定はバスカップラの設定と一致している必要があります。）

Ethernet の場合

バスカプラ型番　　７５０－３４２あるいは７５０－８４２を選択してください。この時点でシリアル通信の各設定条件は無視されます。

IP アドレス　　該当バスカプラの IP アドレスを指定します。

ポート識別番号　　Ethernet の場合は、同時に複数のプログラムを起動して測定することが可能です。起動毎に番号を変えてください。（１～９９まで）

UDP, TCP 通信　　スライドスイッチにて UDP 通信か TCP 通信を選択します。デフォルトは UDP 通信になっています。

シリアル通信の場合

バスカプラ型番　　７５０－３１２／３１４／３１５／３１６／８１２／８１４／８１５／８１６のいずれかを選択します。この時点で Ethernet 通信の各設定条件は無視されます。

ポート番号　　接続した PC の Com ポートの番号を入れます。

パリティチェック　　シリアル通信時に行うパリティチェックを指定します。
0：無し、1：偶数パリティ、2：奇数パリティ *（初期設定では０を入力）*

ボーレート　　▼をクリックすると以下のボーレートのリストが表れるので、バスカップラーに設定されたボーレートの値を選びます。
通常は 9600（初期値）を選択してください。
115200 で動作するパソコンは一般的ですが、安価なパソコン（特にラップトップパソコン）はシリアルポートの設計が簡易化されているものがあります。このような場合は動作に支障をきたすこともありますので、このような場合はボーレートを下げてください。
1200、2400、4800、9600、19200、38400、57600、115200

データビット　　データビット長を指定します。
7：7 ビット長、8：8 ビット長 *（初期設定では８を入力）*

ストップビット　　ストップビット長を指定します。
1：1 ビット長、2：2 ビット長、3：3 ビット長 *（初期設定では１を入力）*

ノードアドレス　　ワゴ I/O のバスカップラーに（ロータリスイッチ）設定されたアドレスと同じ値を選びます。1~99。

エラーチェック　　エラーチェックの有無を指定します。
0：エラーチェックを行いません。
1：エラーチェックを行います。 *（初期設定の場合は１を入力）*

ETHERNET 通信・シリアル通信共通

モジュール型番　構成された I/O モジュールの型番（750XXX）をバスカップラーに近い順に上のボックスから書き込みます。***最後に７５０－６００をタイプします。誤ってボックスを開いてしまったときは、後のボックスにはすべて「７５０－６００」***とタイプしてください。

３－２　ステータス表示

コンフィグステータス　コンフィグが正常に行われたかどうかの確認用です。
0：正常、-1：コンフィグ異常

オープンステータス　シリアルポートが正常にオープンできたかの確認用です。
0：正常、-1：オープン異常

通信テスト結果　I/O との通信中異常が発生したかどうかの確認用として、内部処理において受け取ったコマンドの回数を数えて通信テストチェックをしています。

エラー出力　もし内部処理においてエラーがあった場合、そのエラー情報が表示されます。

３－３　操作ボタン

設定条件を記憶　画面上で設定したパラメータ値を保存用としてファイルに書き込み（記録）します。パラメータ設定を記憶します。***通信処理の実行を押す前に、記憶しておきたい設定パラメータを記憶保存しておきます。***

設定条件を呼出　一度設定パラメータを記憶すると、次回からはファイルに記録されているパラメータの内容を呼び出し、画面上の各ボックスに設定します。

＊置き換えのボタンを押すと条件記憶ファイルが置きかえられます。　条件を記憶するファイルは
　２つありますので「置き換え」を２回クリックしてください。
＊次回からは「設定条件を呼出」を押すと自動的に設定条件が呼び出されます。
＊プログラムメイン画面においても同様の操作で設定条件の記憶と呼出が可能です。

通信処理　実行　前記のコンフィグ設定、通信設定をすべて行った後このボタンを押します。正常に通信処理が行われると WagoPnlConfig1.vi の画面が自動的に閉じて、WAGO_DAQ1.vi の画面が表れます。

エラー情報　クリア　　エラー出力にエラー情報が出た場合は、継続　を押したあと、

このクリアボタンを押して、正しいパラメータをタイプ入力後
再度実行ボタンを押してください。

＜エラーメッセージが発生した場合の処理＞

エラーメッセージが発生した場合は、以下の３とおりの可能性が考えられます。

①バスカプラ型番、コンフィグ（モジュール構成）入力が間違っている。

②IP アドレス、ノードアドレス、あるいはシリアルポート設定が間違っている

③　前回にプログラムを動作させていて、プログラムを強制的に閉じてしまった。

①、②の場合は　OK　を押して、クリアボタンを押した後、正しい情報を再入力してから　実行ボタンを押してください。そのままウィンドウを閉じるとプログラム全体を強制終了させてしまいますので、この場合はすべを閉じて、再度プログラムを立ち上げてください。

③の場合は、ウィンドウズ上で COM ポートが開いたままになっている可能性があります。プログラムを再起動してもエラーが出る場合はウィンドウズを再起動してください。

それでもうまくいかない場合は一度ワゴ I/O システムの電源を入れ直して、プログラムを再度立ち上げてください。

３－４　画面上半分

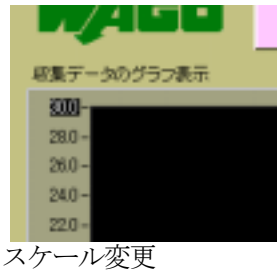
前記の実行ボタンが押されると以降に示した WAGO_DAQ１.vi の画面が表示されます。

以下は WAGO_DAQ３.vi の画面上半分の例を表します。

収集データ　　収集されたデータがプロット表示されます。横軸は収集されるデータ点数を示し、縦軸は入力データの値の大きさを示します。横軸、縦軸共値の変更ができます。表示データは乗算パラメータを乗算した後の結果を表示します。

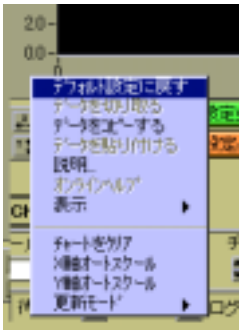
スケール変更　　X 軸および Y 軸のスケール変更するには、両端の数値をクリックして書き換え変更できます。

X 軸のスケール変更



Y 軸のスケール変更

３－５　グラフパラメータの変更



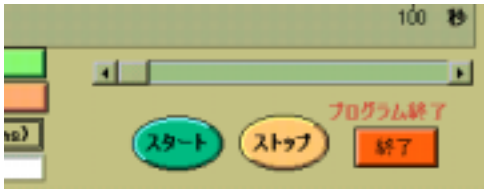
グラフ表示のパラメータ各種を変更する場合は、グラフの原点でマウスを右クリックし、プルダウンメニューを出して変更します。

表示ボックス

現在時刻　　測定時の現在時刻が年/月/日、時：分で表示されます。

CH1～CH8　　1 チャンネルから 8 チャンネルにおける入力データの現在値が表示されます。

スタート、ストップ、プログラム停止



① スタート、ストップ
トリガスイッチが「手動」になっているときは計測のスタートとストップはこのボタンで行います。トリガー計測中に手動で計測をストップさせたいときはトリガースイッチを「手動」にしてからストップボタンを押してください。

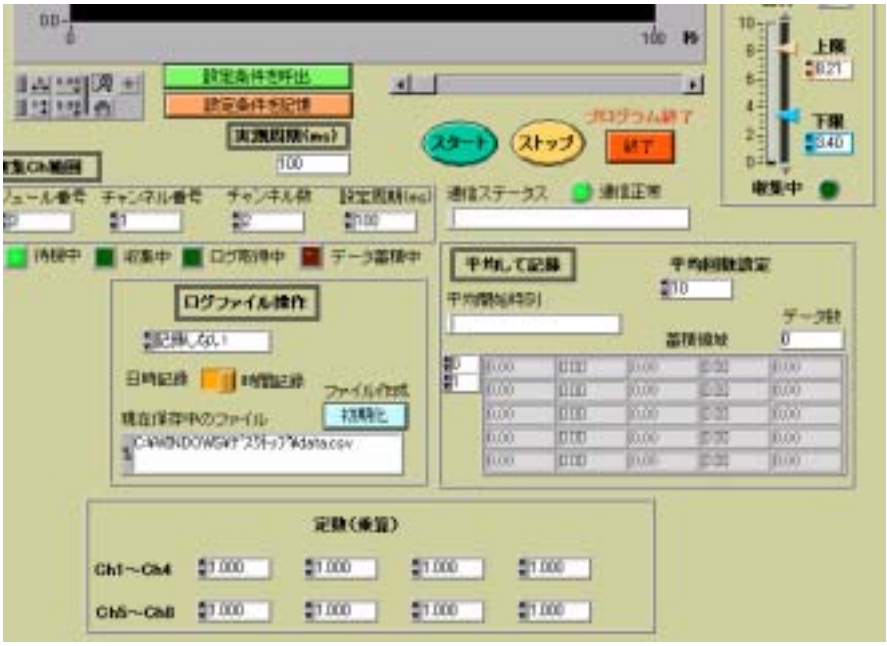
② プログラムすべてを終了させたいときは、プログラム終了ボタンを押してください。以下のように画面左上の「→」が白色に変更になりプログラムがすべて終了します。

＜プログラム動作中＞　　＜プログラム終了中＞

→

再度プログラムを実行させたいときは白色の→をクリックして、ソフトウェアの立ち上げから実行してください。

以下は WAGO_DAQ1.vi の画面の下半分を表します。



収集 Ch 範囲

モジュール番号　　データ収集の対象範囲で開始モジュールの番号を入れます。バスカップラーに一番近いものが 1 となります。

チャンネル番号　　開始モジュール内のデータ収集を開始するチャンネルを入れます。

チャンネル数　　上記の開始チャンネルから同時にデータ収集するチャンネル数を入れます。

＜「収集 Ch を設定してください」のメッセージが表示されたときの処理＞

「収集 Ch 範囲を設定してください」のメッセージが表示されるときは、モジュール番号、チャンネル番号、トリガー部のﾓｼﾞｭｰﾙ番号の設定が正しく行われていないことを意味します。

このときは、モジュール番号に相当するモジュールが出力モジュールになっていないかどうか、チャンネル番号が正しく設定されているかどうか確認・修正してください。

３－６　レンジトリガ

レンジトリガを選択すると、指定されたチャンネルのデータが下限と上限の間に入っているときだけ、自動的に繰り返してデータ収集が可能です。

① 収集 Ch 範囲で設定されたチャンネルの中で監視したいチャンネル番号を入力します。（チャンネルの現在値はグラフの右隣に表示されています。）

② 上限と下限の値を、スライドを動かすかあるいは数値入力で設定します。

③ スライドスイッチをレンジトリガに設定します。

＊このときのデータはすべて電圧、電流、温度などの直接値で設定してください。

＊監視モニタのスケールを変更したいときは一番上の数値を変更してください。

＊レンジトリガでデータ収集中に、マニュアルで待機中に戻したいときは、手動に切り替えてから「データ収集開始／待機」ボタンを押してください。

３－７　ダイナミックトリガ

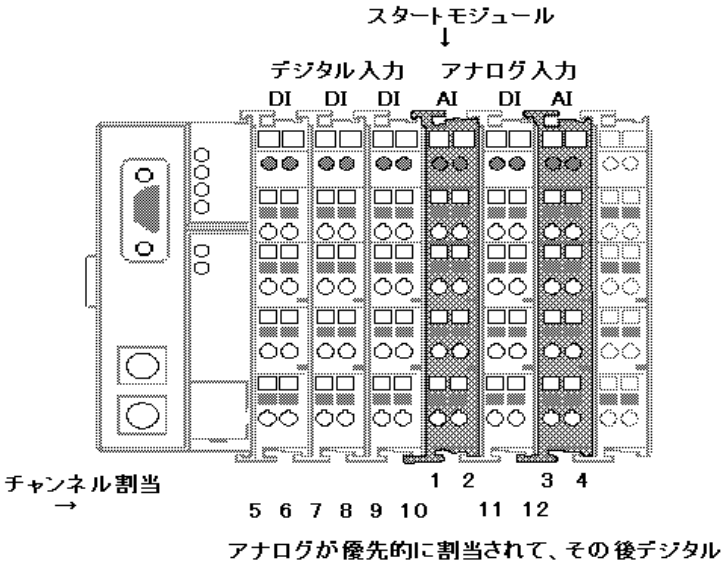
レンジトリガを選択した状態で**監視 Ch に「０」を入力する**とダイナミックトリガ動作を行います。

ダイナミックトリガとは、表示されている８ch 分のデータが一つでも下限値と上限値にあるどきにデータ収集動作を行います。表示８ch 分のデータがすべて下限値と上限値の範囲外になったときに待機状態となります。

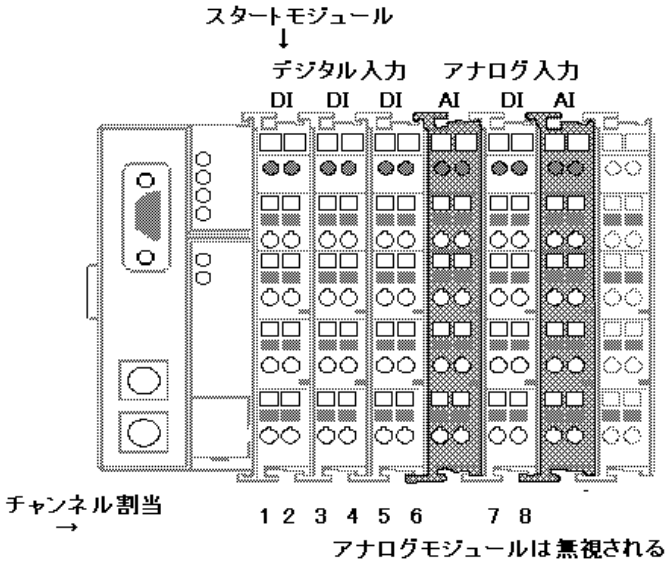
＊表示チャンネルのデータはパラメータ乗算後のデータを表示していますが、トリガーが適用されるデータはパラメータ乗算前の入力データです。パラメータが初期値「1.000」になっているときは表示される値がそのまま適用されます。

３－８　＜モジュール設定とチャンネル割り当てについて＞

＊開始するモジュール番号は入力モジュールになっていなければなりません。開始するモジュールが出力モジュールに設定されていると、データが正しく取得されないことがあります。
＊デジタル、アナログモジュールが混在している場合はスタートチャンネル（モジュール番号およびチャンネル番号）をアナログモジュールから設定してください。
＊デジタルモジュールのデータも「１、０」で取り込めます。デジタルモジュールがアナログモジュールの前（バスカップラに近いほう）に設定されていても、すべてのアナログ入力モジュールの後にデジタルチャンネルが割り当てられます。
（下図はモジュール番号４、チャンネル番号１、チャンネル数１２の場合のチャンネル割り当て例）



＊スタートモジュールをデジタルに設定すると、それ以降のデジタル入力モジュールだけが入力の対象となります。



（上図は、モジュール番号１、チャンネル番号１､チャンネル数８の場合のチャンネル割り当て例）

設定周期(ms)
実測周期(ms)

PC から I/O 全体に対してデータを収集する希望周期を入れます（単位＝ms)。実際に計測されている周期です。***設定周期と同じ常に同じ表示をしている場合には、システム全体が安定したスピードで動作していることを意味します。これはシリアル通信の設定スピード、パソコンの種類などによって変わります。***

設定条件を記憶

画面上で設定したパラメータ値を保存用としてファイルに書き込み（記録）します。パラメータ設定を記憶します。

設定条件を呼出

ファイルに記録されているパラメータの内容を呼び出し、画面上の各ボックスに設定します。

＊ソフトウェアを最初に起動したとき、設定条件を記憶を２回クリックして条件設定記憶ファイルを作成してください。この動作で条件記憶ファイルが自動的に生成されます。この作業を行わずに設定条件を呼出をクリックしてしまうと、すべてのパラメータが「0」に設定されてしまいます。

通信ステータス

「収集 Ch 範囲」においてモジュール番号、チャンネル番号、チャンネル数が正しく設定されていないとき「収集 Ch 範囲を設定してください」のメッセージが表示されます。**このときは、モジュール番号に相当するモジュールが出力モジュールになっていないかどうか、チャンネル番号が正しく設定されているかどうか確認・修正してください。**

表示ランプ

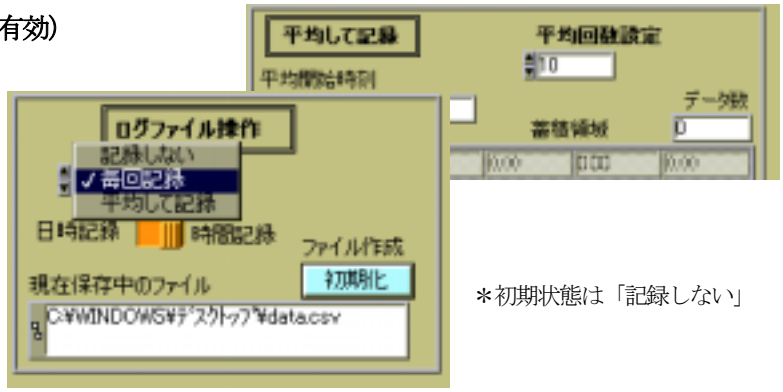
待機中
データ収集が待機中の時に点灯します。

収集中
データ収集が行われている時に点灯します。

ログ取得中
データをファイルに記録中の時に点灯します。

データ蓄積中
平均してデータを記録している時、一時的に蓄積領域に溜めている時に点灯します。

３－９　ログファイル操作（待機中のみ有効）



＊初期状態は「記録しない」

①　毎回記録と平均化処理

計測したデータはサンプリング毎に記録するか、平均化して記録するかが選択できます。

「平均化して記録」を選択したときは「平均回数」を設定してください。例えば１０を入力した場合は１０データを平均して１つのデータとして記録するので、記録されるデータ量は1／１０になります。

①　記録ファイルの初期設定

ファイルの初期化ボタンを押して、データを記録するファイルを選択あるいはファイル名を決めます。ファイル形式はCSVファイルにしてください。ファイルを初期化（オールクリア）しない限りは、新しい行（セル）にデータを追加書き込みしていきます。

＊置き換えボタンを押すと、ファイルがクリアされる。

②　記録されたデータ

	A	B	C	D	E
3	Create Date : 01/01/22 15:30:29				
4	Number of channel	8	Sampling T	300ms	
5	15:31:04	0	0	24.8	850
6	15:31:04	0	0	25	850
7	15:31:05	0	0	25	850
8	15:31:05	0	0	25	850
9	15:31:05	0	0	24.8	850
10	休止				
11	15:31:06	0	0	24.8	850
12	15:31:07	0	0	25	850
13	15:31:07	0	0	25	850
14	休止				
15	平均開始時刻				
16	15:31:15	0	0	24.956	850
17	平均開始時刻				
18	15:31:18	0	0	24.933	850

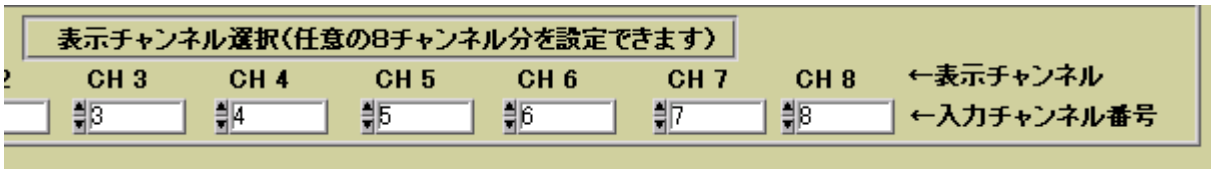
＊データは時刻情報と共に同じ行に記録されます。
＊B列が１Ch目のデータです。
＊データ計測が一時中断した場合は休止の行が挿入されます。
＊「平均して記録」した場合には平均を開始した時刻が記録され、平均したチャンネルごとに記録されます。

３－１０　定数（乗算）

各チャンネルごとのデータにパラメータを乗算したいときにチャンネルごとにパラメータ数値を設定します。初期値は１．０００に設定されています。このパラメータが乗算されたデータはそのままファイルに記録されます。DAQ_Sample4 については乗算パラメータ機能はありません。

＜DAQ_Sample2、2A、2B.exe＞

DAQ_Sample2 においては、入力されている最大 32ch 分のデータから、表示する 8ch 分を選択できます。



「表示チャンネル」：グラフに表示される１CH～８CH を表します。
「入力チャンネル番号」：設定された、入力チャンネルの番号を意味します。最大 32 になりますが、この中から表示したいチャンネルを選んでください。

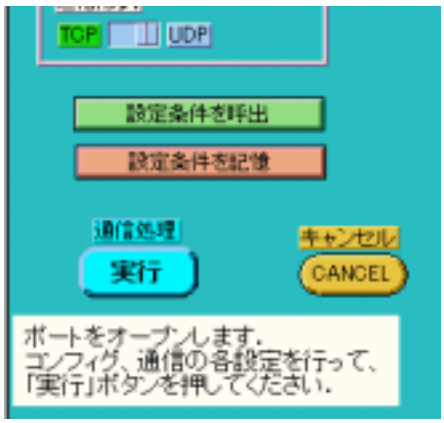
＜DAQ_Sample3、3A、3B.exe＞

DAQ_Sample3 においては、入力されている最大 32ch 分のデータのうち、連続する 8ch ごとに表示を切り替えすることができます。「1ch～8ch」、「9ch～16ch」、「17ch～24ch」、「25ch～32ch」の 4 つのうちから選択できるようにリングメモリになっています。

＜DAQ_Sample4.exe＞

① 複数の子局のオープン

DAQ_Sample4 では同時に４つの子局のデータを収集することができます。最初のノードとの通信を確立した後、２番目～４番目の子局の IP アドレスを指定してポートをオープンします。
途中でポートオープンを中止したいときは　「Cancel」ボタンを押してポートオープン画面を閉じてください。



以下の図のように各々の子局で設定している「チャンネル数」の合計が測定の合計チャンネル数として表示されます。グラフ表示およびエクセルへの記録はこの合計チャンネル数行われます。
測定に関係の無い子局のチャンネル分は「チャンネル数」を「０」に設定すると、その分は自動的に無視されます。

＊現在通信中の子局は「通信」部分が点灯します。
＊現在測定中の子局のチャンネル部分は「入力状態」が点灯します。

② チャンネルの表示

DAQ_Sample４においては、入力されている最大160ch分のデータのうち、連続する8ch ごとに表示を切り替えすることができます。「1ch～8ch」、「9ch～16ch」、「17ch～24ch」・・・「153ch～160ch」のうちから 選択できるようにリングメモリになっています。

４．　アラーム警報出力＜DAQ__Sample1A,　１B,　2A、　2B、3A、3B＞

監視したいチャンネルを指定して、上限値及び下限値を設定してデータがその範囲内に入ったときに警報出力を出力することができます。

メイン画面の「出力条件設定」ボタンを押すと上のような画面が現れます。

①監視 Ch は、メイン表示されている８Ch のうちから選択します。（１～８）
　監視チャンネルを選択すると現在の値がバーグラフで表示されます。
②上限値と下限値を数値入力するか、指示計を動かして設定します。
③バーグラフスケール表示の上下の値をクリックして変更してスケーリング変更できます。
④出力モジュールの番号（バスカップラに近い方から数える）とそのモジュール内でのチャンネル番号を指定します。(デジタル出力モジュールのみ設定可)
⑤動作ボタンを ON にします。
⑥設定した出力条件を記憶させておきたいときは、設定条件を記憶を押して、条件記憶ファイルを置換します。
⑦設定を完了したら、完了ボタンを押して画面を閉じます。

＊*ソフトウェアを最初に起動したとき、設定条件を記憶を２回クリックして条件設定記憶ファイルを作成してください。この動作で条件記憶ファイルが自動的に生成されます。この作業を行わずに設定条件を呼出をクリックしてしまうと、すべてのパラメータが「0」に設定されてしまいます。*

＊*監視 Ch を選択すると現在の値がリアルタイムで表示されます。このデータはメイン画面で表示されているデータ、すなわち乗算パラメータが乗算された後のデータです。*
＊*出力モジュールおよび出力チャンネルは同じチャンネルを同時に指定しないでください。出力動作がうまくいかない場合があります。*
＊*動作ボタンを ON にしないと動作は行われません。*

設定を完了するときは必ず完了ボタンで終了してください。×でウィンドウを

閉じてしまうとプログラム全体を終了してしまいますので、プログラム立ち上げからやり直さなければなりません。

設定画面を閉じたら、「待機中も警報出力」するか、「データ収集中だけ警報出力するかを選択します。

5． タイマー繰り返し計測＜DAQ__Sample１B， 2B、3B＞

トリガーによる計測ではなくて、タイマーで開始時刻と計測時間、繰り返し回数を設定して、自動計測することが可能です。タイマー設定ボタンを押すと以下のような設定画面が現れます。

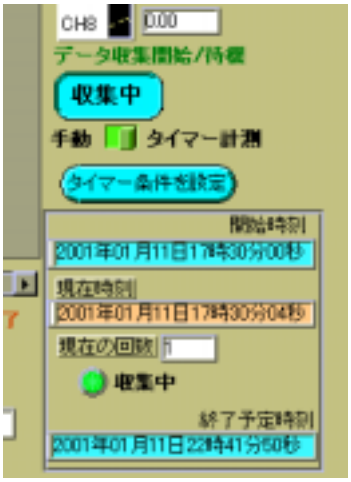
測定開始時刻、およびデータ収集時間と休止時間、繰り返し回数を数値入力した後、完了ボタンを押して画面を終了させてメイン画面に戻ります。

＊*設定を完了するときは必ず完了ボタンで終了してください。*X*でウィンドウを閉じてしまうとプログラム全体を終了してしまいますので、プログラム立ち上げからやり直さなければなりません。*

タイマー設定画面を完了させた後、タイマー計測スイッチを ON にします。

このスイッチを ON にしないと、開始時刻になっても動作しません。

画面には現在の時刻のリアルタイム表示、開始時刻、計測中の現在の繰り返し回数、および終了予定時刻が表示されます。



以上

取扱い説明書　バージョン　7.0　(2007 年 9 月現在)

ワゴジャパン株式会社　営業部オートメーション
〒136-0071 東京都江東区亀戸 1-5-7 日鐵 ND タワー
TEL:03-5627-2059　FAX:03-5627-2055